International Journal on Advances in Intelligent Systems

2014 vol. 7 nr. 1&2

The *International Journal on Advances in Intelligent Systems* is Published by IARIA.
ISSN: 1942-2679
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Intelligent Systems
Volume 7, Number 1 & 2, 2014

Phan Cong-Vinh, NTT University, Vietnam
Christophe Cruz, University of Bourgogne, France
Beata Czarnacka-Chrobot, Warsaw School of Economics, Department of Business Informatics, Poland
Claudia d'Amato, University of Bari, Italy
Sérgio Roberto P. da Silva, Universidade Estadual de Maringá - Paraná, Brazil
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Dragos Datcu, Netherlands Defense Academy / Delft University of Technology , The Netherlands
Antonio De Nicola, ENEA, Italy
Claudio de Castro Monteiro, Federal Institute of Education, Science and Technology of Tocantins, Brazil
Noel De Palma, Joseph Fourier University, France
Jan Dedek, Charles University in Prague, Czech Republic
Zhi-Hong Deng, Peking University, China
Stojan Denic, Toshiba Research Europe Limited, UK
Vivek S. Deshpande, MIT College of Engineering - Pune, India
Sotirios Ch. Diamantas, Pusan National University, South Korea
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Jerome Dinet, Univeristé Paul Verlaine - Metz, France
Jianguo Ding, University of Luxembourg, Luxembourg
Yulin Ding, Defence Science & Technology Organisation Edinburgh, Australia
Alexiei Dingli, University of Malta, Malta
Mihaela Dinsoreanu, Technical University of Cluj-Napoca, Romania
Ioanna Dionysiou, University of Nicosia, Cyprus
Roland Dodd, CQUniversity, Australia
Nima Dokoohaki, Royal Institute of Technology (KTH)-Kista, Sweden
Suzana Dragicevic, Simon Fraser University- Burnaby, Canada
Mauro Dragone, University College Dublin (UCD), Ireland
Marek J. Druzdzel, University of Pittsburgh, USA
Carlos Duarte, University of Lisbon, Portugal
Raimund K. Ege, Northern Illinois University, USA
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Larbi Esmahi, Athabasca University, Canada
Simon G. Fabri, University of Malta, Malta
Umar Farooq, Amazon.com, USA
Mehdi Farshbaf-Sahih-Sorkhabi, Azad University - Tehran / Fanavaran co., Tehran, Iran
Anna Fensel, Semantic Technology Institute (STI) Innsbruck and FTW Forschungszentrum Telekommunikation Wien, Austria
Stenio Fernandes, Federal University of Pernambuco (CIn/UFPE), Brazil
Oscar Ferrandez Escamez, University of Utah, USA
Florin Filip, Romanian Academy, Romania
Agata Filipowska, Poznan University of Economics, Poland
Ziny Flikop, Scientist, USA
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Francesco Fontanella, University of Cassino and Southern Lazio, Italy
Panagiotis Fotaris, University of Macedonia, Greece
Enrico Francesconi, ITTIG - CNR / Institute of Legal Information Theory and Techniques / Italian National Research Council, Italy

Rita Francese, Università di Salerno - Fisciano, Italy
Bernhard Freudenthaler, Software Competence Center Hagenberg GmbH, Austria
Sören Frey, syscovery Business Solutions GmbH, Germany
Steffen Fries, Siemens AG, Corporate Technology - Munich, Germany
Somchart Fugkeaw, Thai Digital ID Co., Ltd., Thailand
Naoki Fukuta, Shizuoka University, Japan
Mathias Funk, Eindhoven University of Technology, The Netherlands
Adam M. Gadomski**,** Università degli Studi di Roma La Sapienza, Italy
Alex Galis, University College London (UCL), UK
Crescenzio Gallo, Department of Clinical and Experimental Medicine - University of Foggia, Italy
Matjaz Gams, Jozef Stefan Institute-Ljubljana, Slovenia
Raúl García Castro, Universidad Politécnica de Madrid, Spain
Fabio Gasparetti, Roma Tre University - Artificial Intelligence Lab, Italy
Joseph A. Giampapa, Carnegie Mellon University, USA
George Giannakopoulos, NCSR Demokritos, Greece
David Gil, University of Alicante, Spain
Harald Gjermundrod, University of Nicosia, Cyprus
Angelantonio Gnazzo, Telecom Italia - Torino, Italy
Luis Gomes, Universidade Nova Lisboa, Portugal
Nan-Wei Gong, MIT Media Laboratory, USA
Francisco Alejandro Gonzale-Horta, National Institute for Astrophysics, Optics, and Electronics (INAOE), Mexico
Sotirios K. Goudos, Aristotle University of Thessaloniki, Greece
Victor Govindaswamy, Texas A&M University-Texarkana, USA
Gregor Grambow, University of Ulm, Germany
Fabio Grandi, University of Bologna, Italy
Andrina Granić, University of Split, Croatia
Carmine Gravino, Università degli Studi di Salerno, Italy
Dominic Greenwood, Whitestein Technologies, Switzerland
Michael Grottke, University of Erlangen-Nuremberg, Germany
Vic Grout, Glyndŵr University, UK
Maik Günther, Stadtwerke München GmbH, Germany
Francesco Guerra, University of Modena and Reggio Emilia, Italy
Alessio Gugliotta, Innova SPA, Italy
Richard Gunstone, Bournemouth University, UK
Fikret Gurgen, Bogazici University, Turkey
Maki Habib, The American University in Cairo, Egypt
Till Halbach Røssvoll, Norwegian Computing Center, Norway
Jameleddine Hassine, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia
Ourania Hatzi, Harokopio University of Athens, Greece
Yulan He, Aston University, UK
Kari Heikkinen, Lappeenranta University of Technology, Finland
Cory Henson, Wright State University / Kno.e.sis Center, USA
Arthur Herzog, Technische Universität Darmstadt, Germany
Rattikorn Hewett, Whitacre College of Engineering, Texas Tech University, USA
Celso Massaki Hirata, Instituto Tecnológico de Aeronáutica - São José dos Campos, Brazil
Jochen Hirth, University of Kaiserslautern, Germany

Susana Munoz Hernández, Universidad Politécnica de Madrid, Spain
Adrian Muscat, University of Malta, Malta
Peter Mutschke, GESIS - Leibniz Institute for the Social Sciences - Bonn, Germany
Bela Mutschler, Hochschule Ravensburg-Weingarten, Germany
Deok Hee Nam, Wilberforce University, USA
Fazel Naghdy, University of Wollongong, Australia
Joan Navarro, Research Group in Distributed Systems (La Salle - Ramon Llull University), Spain
Saša Nešić, University of Lugano, Switzerland
Rui Neves Madeira, Instituto Politécnico de Setúbal / Universidade Nova de Lisboa, Portugal
Toàn Nguyên, INRIA Grenobel Rhone-Alpes/ Montbonnot, France
Andrzej Niesler, Institute of Business Informatics, Wroclaw University of Economics, Poland
Michael P. Oakes, University of Sunderland, UK
John O'Donovan, University of California - Santa Barbara, USA
Kouzou Ohara, Aoyama Gakuin University, Japan
Jonice Oliveira, Universidade Federal do Rio de Janeiro, Brazil
Ian Oliver, Nokia Location & Commerce, Finland / University of Brighton, UK
Michael Adeyeye Oluwasegun, University of Cape Town, South Africa
Sigeru Omatu, Osaka Institute of Technology, Japan
Sascha Opletal, University of Stuttgart, Germany
Flavio Oquendo, European University of Brittany/IRISA-UBS, France
Fakri Othman, Cardiff Metropolitan University, UK
Enn Õunapuu, Tallinn University of Technology, Estonia
Jeffrey Junfeng Pan, Facebook Inc., USA
Hervé Panetto, University of Lorraine, France
Malgorzata Pankowska, University of Economics, Poland
Harris Papadopoulos, Frederick University, Cyprus
Laura Papaleo, ICT Department - Province of Genoa & University of Genoa, Italy
Agis Papantoniou, National Technical University of Athens, Greece
Thanasis G. Papaioannou, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Andreas Papasalouros, University of the Aegean, Greece
Eric Paquet, National Research Council / University of Ottawa, Canada
Kunal Patel, Ingenuity Systems, USA
Carlos Pedrinaci, Knowledge Media Institute, The Open University, UK
Juan C Pelaez, Defense Information Systems Agency, USA
Yoseba Penya, University of Deusto - DeustoTech (Basque Country), Spain
Cathryn Peoples, University of Ulster, UK
Asier Perallos, University of Deusto, Spain
Christian Percebois, Université Paul Sabatier - IRIT, France
Andrea Perego, European Commission, Joint Research Centre, Italy
Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada
Willy Picard, Poznań University of Economics, Poland
Meikel Poess, Oracle, USA
Agostino Poggi, Università degli Studi di Parma, Italy
R. Ponnusamy, Madha Engineering College-Anna University, India
Dorin Popescu, University of Craiova, Romania
Stefan Poslad, Queen Mary University of London, UK

Wendy Powley, Queen's University, Canada
Radu-Emil Precup, "Politehnica" University of Timisoara, Romania
Jerzy Prekurat, Canadian Bank Note Co. Ltd., Canada
Didier Puzenat, Université des Antilles et de la Guyane, France
Sita Ramakrishnan, Monash University, Australia
Elmano Ramalho Cavalcanti, Federal University of Campina Grande, Brazil
Juwel Rana, Luleå University of Technology, Sweden
Martin Randles, School of Computing and Mathematical Sciences, Liverpool John Moores University, UK
Christoph Rasche, University of Paderborn, Germany
Ann Reddipogu, ManyWorlds UK Ltd, UK
Ramana Reddy, West Virginia University, USA
René Reiners, Fraunhofer FIT - Sankt Augustin, Germany
Paolo Remagnino, Kingston University - Surrey, UK
Sebastian Rieger, University of Applied Sciences Fulda, Germany
Andreas Riener, Johannes Kepler University Linz, Austria
Ivan Rodero, NSF Center for Autonomic Computing, Rutgers University - Piscataway, USA
Joel Rodrigues, Instituto de Telecomunicações / University of Beira Interior, Portugal
Alejandro Rodríguez González, University Carlos III of Madrid, Spain
Aitor Rodríguez-Alsina, University Autonoma of Barcelona (UAB), Spain
Paolo Romano, INESC-ID Lisbon, Portugal
Vicente-Arturo Romero-Zaldivar, Atos Origin SAE, Spain
Agostinho Rosa, Instituto de Sistemas e Robótica, Portugal
José Rouillard, University of Lille, France
Paweł Różycki, University of Information Technology and Management (UITM) in Rzeszów, Poland
Igor Ruiz-Agundez, DeustoTech, University of Deusto, Spain
Michele Ruta, Politecnico di Bari, Italy
Melike Sah, Trinity College Dublin, Ireland
Francesc Saigi Rubió, Universitat Oberta de Catalunya, Spain
Abdel-Badeeh M. Salem, Ain Shams University, Egypt
Yacine Sam, Université François-Rabelais Tours, France
Ismael Sanz, Universitat Jaume I, Spain
Ricardo Sanz, Universidad Politecnica de Madrid, Spain
Marcello Sarini, Università degli Studi Milano-Bicocca - Milano, Italy
Munehiko Sasajima, I.S.I.R., Osaka University, Japan
Minoru Sasaki, Ibaraki University, Japan
Hiroyuki Sato, University of Tokyo, Japan
Jürgen Sauer, Universität Oldenburg, Germany
Patrick Sayd, CEA List, France
Dominique Scapin, INRIA - Le Chesnay, France
Kenneth Scerri, University of Malta, Malta
Adriana Schiopoiu Burlea, University of Craiova, Romania
Rainer Schmidt, Austrian Institute of Technology, Austria
Bruno Schulze, National Laboratory for Scientific Computing - LNCC, Brazil
Wieland Schwinger, Johannes Kepler University Linz, Austria
Hans-Werner Sehring, T-Systems Multimedia Solutions GmbH, Germany
Paulo Jorge Sequeira Gonçalves, Polytechnic Institute of Castelo Branco, Portugal

Sandra Sendra Compte, Polytechnic University of Valencia, Spain
Kewei Sha, Oklahoma City University, USA
Hossein Sharif, University of Portsmouth, UK
Roman Y. Shtykh, Rakuten, Inc., Japan
Kwang Mong Sim, Gwangju Insitute of Science & Technology, South Korea
Robin JS Sloan, University of Abertay Dundee, UK
Vasco N. G. J. Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Don Sofge, Naval Research Laboratory, USA
Christoph Sondermann-Woelke, Universitaet Paderborn, Germany
George Spanoudakis, City University London, UK
Vladimir Stantchev, SRH University Berlin, Germany
Claudius Stern, University of Paderborn, Germany
Mari Carmen Suárez-Figueroa, Universidad Politécnica de Madrid (UPM), Spain
Kåre Synnes, Luleå University of Technology, Sweden
Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland
Yehia Taher, ERISS - Tilburg University, The Netherlands
Yutaka Takahashi, Senshu University, Japan
Azzelarabe Taleb-Bendiab, Liverpool John Moores University, UK
Dan Tamir, Texas State University, USA
Jinhui Tang, Nanjing University of Science and Technology, P.R. China
Yi Tang, Chinese Academy of Sciences, China
Said Tazi, LAAS-CNRS, Universite Toulouse 1, France
John Terzakis, Intel, USA
Sotirios Terzis, University of Strathclyde, UK
Vagan Terziyan, University of Jyvaskyla, Finland
Ioan Toma, STI Innsbruck/University Innsbruck, Austria
Lucio Tommaso De Paolis, Department of Innovation Engineering - University of Salento, Italy
Davide Tosi, Università degli Studi dell'Insubria, Italy
Raquel Trillo Lado, University of Zaragoza, Spain
Tuan Anh Trinh, Budapest University of Technology and Economics, Hungary
Simon Tsang, Applied Communication Sciences, USA
Theodore Tsiligiridis, Agricultural University of Athens, Greece
Antonios Tsourdos, Cranfield University, UK
José Valente de Oliveira, University of Algarve, Portugal
Cristián Felipe Varas Schuda, NIC Chile Research Labs, Chile
Eugen Volk, University of Stuttgart, Germany
Mihaela Vranić, University of Zagreb, Croatia
Chieh-Yih Wan, Intel Labs, Intel Corporation, USA
Jue Wang, Washington University in St. Louis, USA
Shenghui Wang, OCLC Leiden, The Netherlands
Zhonglei Wang, Karlsruhe Institute of Technology (KIT), Germany
Laurent Wendling, University Descartes (Paris 5), France
Maarten Weyn, University of Antwerp, Belgium
Nancy Wiegand, University of Wisconsin-Madison, USA
Alexander Wijesinha, Towson University, USA

International Journal on Advances in Intelligent Systems
Volume 7, Numbers 1 & 2, 2014

CONTENTS

myIdP - The Personal Attribute Hub: Prototype and Quality of Claims

Annett Laube and Severin Hauser
Institute of ICT-based Management
Bern University of Applied Sciences
Biel/Bienne, Switzerland
Email: annett.laube@bfh.ch, severin.hauser@bfh.ch

***Abstract*—The myIdP service is an extension to the Swiss eID infrastructure with the aim to provide a service that handles personal attributes (like address, telephone number, email), which are neither part of the SuisseID identity providers nor of a Claim Assertion Service (CAS), because there is no official authority owning and certifying these data. The myIdP service is a CAS that can reuse data which a user has already given to an application via an Internet transaction. The data is thus validated by the web application before being transferred - as Security Assertion Markup Language (SAML) 2.0 attribute assertion - to the myIdP service. The myIdP service comes in two flavors with different trust relations: the attribute provider and the claim proxy. The attribute provider unites several claims for a given attribute and provides an optional quality assessment before sending it to a requesting web application. A trust relationship must consist between myIdP and the web application. The claim proxy only collects the received claims for a given attribute and transfers them with all details to the requesting application. The application can evaluate the confidence in the data based on the claim details. The model to assess the quality and trustworthiness of the data is based mainly on three factors: freshness of information, quality of the attribute issuer and recurrence of information. The myIdP service is evaluated in a scenario of prefilling e-forms in an eGovernment application.**

***Keywords*-electronic identity, SuisseID, attribute authority, e-form, quality assessment.**

I. Introduction

In a previous paper [1], we presented myIdP, a service based on the SuisseID Infrastructure, an infrastructure for electronic proof of identities (eID) in Switzerland introduced in 2012. In this extended version of the paper we describe the prototype in more detail and discuss the quality model for claims.

The basis of the SuisseID Infrastructure is the SuisseID [2], which is available as USB stick or chip card and contains two digital certificates: (1) the SuisseID identification and authentication certificate (IAC) and (2) the SuisseID qualified digital certificate (QC). The SuisseID IAC can be used to identify the owner in Internet transactions. The SuisseID QC can be used to sign electronic documents in a forgery-proof manner and is not used in the context of myIdP. The SuisseIDs are issued by identity providers (IdP). Contrary to other European countries, where electronic identities are issued by the government together with offline identification (ID card), there are one governmental, but two commercial SuisseID IdPs [3][4] at present.

The SuisseID and its certificates contain only a minimum of personal data (SuisseID number, name or pseudonym and optional email address), due to stringent privacy and data protection requirements in Switzerland. Additionally, a subset of the personal data from the identification document (e.g., a passport) and a well-defined set of additional attributes gathered during the registration process (so called registration process data, RPD) are stored in the identity provider service (extended IdP). The only way to retrieve this data is by strong authentication with the IdP service, using the appropriate SuisseID IAC. The SuisseID Infrastructure is completed with a set of Claim Assertion Services (CAS) [5]. The purpose of a CAS is to provide and certify specific properties or attributes, which had been assigned to the SuisseID owner by some private or public authority. Examples are the membership of an organization or a company, and the proof of professional qualifications, like a notary or a doctor. Especially in the context of eGovernment there is a need for an extension of the beforehand described SuisseID Infrastructure.

More personal attributes (like invoice address, telephone number, email) used in web applications, e.g., online shops, or in electronic forms often used in the eGovernment, are neither subject of the SuisseID IdPs nor the CAS, because there is no official authority owning and certifying these data. The myIdP service fills that gap and allows a SuisseID owner to store and maintain personal attributes. The idea is to store information, which was at least entered (and thus used) once in a web application, for later reuse. The data is used and thus validated by the web application before being transferred as Security Assertion Markup Language (SAML) attribute statement [6] (the so-called attribute claim) to the myIdP service. After that, the user can reuse the attribute for other applications, which improves usability and reduces the error potential in the daily internet transactions.

This paper starts with the related work in Section II, then outlines the architecture, components and flavors of myIdP in Section III. Privacy and data security are subjects of Section IV. In Section V, the integration of myIdP in a scenario of prefilling e-forms is shown. The myIdP quality assessment and trustworthiness is discussed in Section VI. Section VII concludes the document and gives an outlook on further improvements of myIdP.

II. Related Work

A service like myIdP or a SuisseID CAS technically corresponds to an Attribute Authority defined by SAML [6]: An Attribute Authority is a system entity that produces attribute assertions [7].

In general, most of the known SAML Identity Providers (IdPs) can act as an Attribute Authority and issue attribute assertions beside their usual authentication functionality. Examples are the government-issued electronic identities of the European Countries, like the German Identity Card [8], the beID from Belgium [9] or the Citizens Card from Austria [10]. Similar to the SuisseID, all these government-issued eIDs provide only a small number of personal attributes related to the identity document they belong to.

The national electronic identities of the European member states are made interoperable with the STORK European eID Interoperability Platform [11]. With six pilots, the STORK project offers several cross-border eGovernment identity services. In the follow-up project STORK 2.0 [12] also personal attributes related to eIDs are subject of investigation. E.g., in the banking pilot, public and private identity and attribute providers are included in the process of ”Opening a bank account” in a foreign country, online, with a national eID, without physical presence. myIdP could be used in this context as attribute provider for personal attributes, like address, telephone number, email, etc.

In contrast to the central, government-regulated eID services, OpenID [13] is a decentralized authentification service for web based services. The user is free to choose his favourite OpenID identity provider to get an OpenID, which is an URL or XRI including an end-user chosen name (e.g., alice.openid.example.org). OpenID providers are, e.g., Clavid [14], CloudID [15], Google [16] etc. The OpenID providers themselves can support different authentification methods. For example, Clavid offers username/password, one time passwords, SuisseID authentication and the biometric AXSionics Internet Passport.

User attributes, like name, gender or favorite movies, can also be transferred from the OpenID identity providers to the relying party following the OpenID Attribute Exchange Specification [17]. The attributes can be defined (almost) freely, according to the requirements of the relying party. As many OpenID providers do not validate the information entered by the users, the provided attributes have a low level of assurance. There is a need for a validation by a trusted 3rd party, a so called attribute provider (AP). Google started the Open Attribute Exchange Network (Open AXN, also known as ”street identity”, see [18]) to include validated information from APs. myIdP has the potential to act as an OpenID attribute provider, but is currently only enabled to be used together with the SuisseID.

WebIDs [19] are especially common in social media (Facebook, LinkedIn, etc.) to allow users to identify themselves in order to publish information. Each user can make their own WebID or rely on an identity provider. The WebID is a URL with a #tag pointing to a FOAF file [20] that contains a cross-link to a (self-generated) certificate. Information that should be included but is not required in a WebID Profile are the name (foaf:name) of the individual or agent, the email address associated (foaf:mbox) and the agent’s image (foaf:depiction). More attributes and links to all kind of web objects (other persons, groups, publications, account, etc.) can be included as well. WebIDs can be connected to OpenIDs and vice versa.

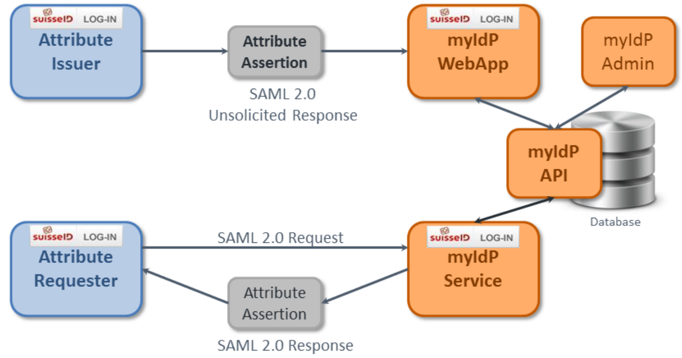


Figure 1. myIdP components and service provider roles

III. Architecture

myIdP consists of four components (see orange boxes in Figure 1): the myIdP Service, the myIdP WebApp, the myIdP Admin and the myIdP API.

A. *myIdP Service*

The **myIdP Service** is an attribute authority according to the SAML 2.0 standard [6] distributing assertions in response to identity attribute queries from an entity acting as an attribute requester. Like a typical SuisseID CAS [5], the users can – after a successful authentication with their SuisseID IAC – select and confirm attributes, which were formerly received from an attribute issuer, e.g., a web shop, and are stored in the myIdP database. The available attributes are not fixed. They depend on the application scenario and can be configured with the help of the myIdP Admin tool.

New to the concept of CAS is the provisioning of a quality (level of assurance, level of confidence) together within the attribute assertions. myIdP integrates a quality module that calculates the trustworthiness of the provided information on the basis of the age, number of affirmations and quality of the issuer of the received and stored attribute assertions. This assurance level or quality can be used by an attribute requester to insist on a certain level of assurance for the requested attributes. The different approaches to calculate the assurance level or quality are discussed in Section VI.

The myIdP Service is available in two flavors: the **Attribute Provider** and the **Claim Proxy**.

The Attribute Provider summarizes the attribute assertions available in the myIdP database for the given request. All details about the original attribute providers of the information are hidden. After the user has selected and confirmed the attribute values, the newly built attribute assertion is signed by the myIdP Service. When requested, an assurance level is included in this assertion. For this myIdP flavor, a direct trust relationship is established between the myIdP Service and the web application in the attribute requester role.

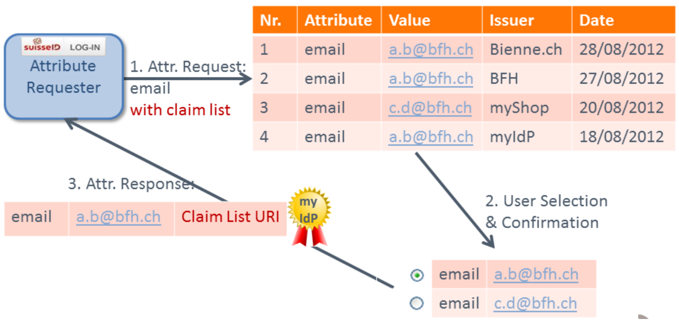


Figure 2. Example Processing Attribute Request

In Figure 2, an example of the process of constructing a response for an attribute request is shown. The attribute *email* is requested. In the myIdP database, four records with two different values are found. The two values are displayed to the user, who selects and confirms which one to use. The SAML attribute response then contains this value with myIdP as attribute issuer.

This is different in the second myIdP flavor. The Claim Proxy extracts the stored attribute assertions from the myIdP database for a given attribute request. After the selection of attribute values and the explicit confirmation by the user, an attribute assertion containing an URI and optionally the assurance level is returned to the requesting web application (the differences between the Attribute Provider and Proxy Mode are highlighted in red in Figure 2). This attribute request is also signed by myIdP but only to ensure integrity. The web application can use the URI from the attribute assertion to assess the originally received attribute assertions enveloped in an XML document. After downloading the XML document, the web application can access all details of the original assertions, including the issuers and timestamps, and perform its own quality assessment. The trust relationship has changed: the web application trusts the attribute issuers directly.

In order to support the provision of a quality assessment of an attribute value and of the claim list URI, the SAML attribute assertions was extended (see the XSD fragment shown in Figure 3).

The SAML attribute request contains an additional flag (attribute *ClaimList*), that indicates the use of claim proxy mode. The SAML attribute response contains as result an URI pointing to a list of claims (attribute assertions) extracted from the myIdP database (attribute *ClaimListURI*). See the shortened example of a SAML attribute response in Figure 4. The URI is only for a short time available to the service provider and enables the service provider to download all claim details.

An example for the provided claimlists is shown conceptionally in Figure 5. Corresponding to the example used in Figure

```
<complexType name="AttributeType">
  <sequence>
    <element ref="saml:AttributeValue"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Name" type="string"
    use="required" />
  <attribute name="NameFormat" type="anyURI"
    use="optional" />
  <attribute name="FriendlyName" type="string"
    use="optional" />
  <attribute name="myidp:Quality" type="decimal"
    use="optional" />
  <attribute name="myidp:ClaimListURI" type="anyURI"
    use="optional" />
  <attribute name="myidp:ClaimList" type="boolean"
    use="optional" />
  <anyAttribute namespace="##other"
    processContents="lax" />
</complexType>
```

Figure 3. Extended xsd `AttributeType`

```
<saml2p:Response>
 <saml2:Issuer>https://myidp.bfh.ch:8443
 </saml2:Issuer>
 <ds:Signature> ... </ds:Signature>
 <saml2p:Status>
  <saml2p:StatusCode Value=
    "urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </saml2p:Status>
 <saml2:Assertion>
  <saml2:Issuer>https://myidp.bfh.ch:8443
        </saml2:Issuer>
  <ds:Signature> ... </ds:Signature>
  <saml2:Subject>
   <saml2:NameID>1300-0000-0001-0001</saml2:NameID>
    ...
  </saml2:Subject>
  <saml2:Conditions
     NotBefore="2014-01-17T13:12:23.922Z"
     NotOnOrAfter="2014-01-17T13:22:25.922Z"/>
  <saml2:AttributeStatement>
   <saml2:Attribute
     Name="http://www.ech.ch/xmlns/
                 eCH-0046/2/emailAddress"
     NameFormat="urn:oasis:names:tc:SAML:2.0:
                       attrname-format:uri"
     myidp:issuerlisturi="https://myidp.bfh.ch:8443/
           myidp-service/requestXml?param=397332808"
     xmlns:myidp=
           "http://www.myidp.ch/xmlns/schema/v1">
    <saml2:AttributeValue>a.b@bfh.ch
    </saml2:AttributeValue>
   </saml2:Attribute>
  </saml2:AttributeStatement>
 </saml2:Assertion>
</saml2p:Response>
```

Figure 4. SAML Attribute response

2, the claim list contains three entries. The service provider can use the information to do its own quality assessment.

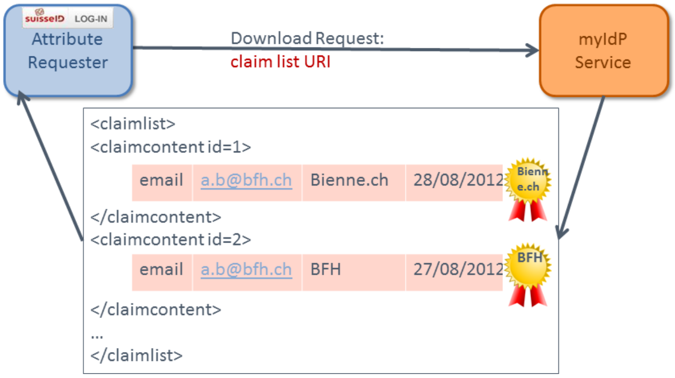


Figure 5. Example Claim Proxy - Claim List

B. myIdP WebApp

The **myIdP WebApp** is the end user front-end of myIdP, where users – after the successful authentication with their SuisseID IAC – can view and manage their attributes. Attributes cannot be entered directly into the myIdP WebApp, except the master data related to the myIdP account (billing address and contact email). Attributes always come from a service provider, e.g., a web application, acting as attribute issuer, which forwards – after confirmation by the user – the attribute assertions to myIdP. The attributes then arrive in the so-called **Inbox** (see Figure 7), where they can be viewed in detail and manually activated before they are exposed via the myIdP service. Corresponding to the user-centric approach of the SuisseID, the users are always in control of their data and can activate/deactivate and delete attributes at any time. As a side effect, the user gets a usage history of their attributes in the web.

C. myIdP Admin

The **myIdP Admin** is an administration tool for myIdP. It supports the maintenance of attribute definitions and the registration process of service providers, which is needed to set up secure connections and trust relationships. A service provider can register as attribute issuer or attribute requester (see Section III-E). The registration details contain the used URLs, certificates and optional SAML metadata.

The registration also allows myIdP to ensure that the service providers well behave. Otherwise one might for instance issue non-consented claims.

New attributes can be enabled for usage within the myIdP community simply by importing the related XML Schemata or by using the SAML Metadata Exchange [21].

In order to assist the research activities to develop an appropriate quality model, the myIdP Admin supports the substitution of the used quality model. Due to the use of the Strategy design pattern, it is possible to develop and integrate new quality calculation models at any time.

D. myIdP API

The **myIdP API** provides an interface to the central database commonly used by the other three myIdP components.

E. Service Providers

A service provider can interact with myIdP, incorporating two roles (see the blue boxes in Figure 1):

- **Attribute Requester:** The service provider electronically sends an attribute query to the myIdP Service in order to draw a confirmation statement - a SAML 2.0 attribute assertion - from the myIdP service and uses it in further actions, e.g., prefilling of web forms.
- **Attribute Issuer:** The service provider sends SAML 2.0 attribute statements (unsolicited SAML response) to myIdP. (Despite the possibility to group several attributes in one SAML statement, myIdP prefers single attribute statements, in order to expose a minimum of information in the claim proxy case.) The attribute values were entered either manually into the web application by the users or have been requested beforehand from the myIdP Service.
  A special attribute issuer is the myIdP WebApp, which uses the master data (address, email) entered during the myIdP registration process, to provide the first attribute statements to the users.

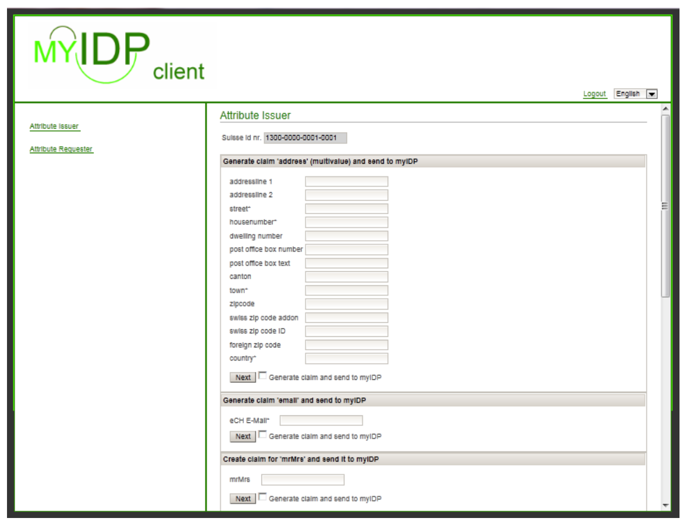


Figure 6. Screenshot myIdP Client - Attribute Issuer

To demonstrate and test the behaviour of a service provider, a demo web application, the **myIdP Client**, was developed. The myIdP Client offers two functionalities, corresponding to the two service provider roles: sending attributes to the myIdP Service and requesting attributes.

In Figure 6, the screen for acting as an attribute issuer is shown. The user can, like in a normal web application, enter some data, e.g., an address or the email. When the user hits the button ”next” and has checked the box beside, to confirm the disclosure of his data to myIdP, the myIdP client sends

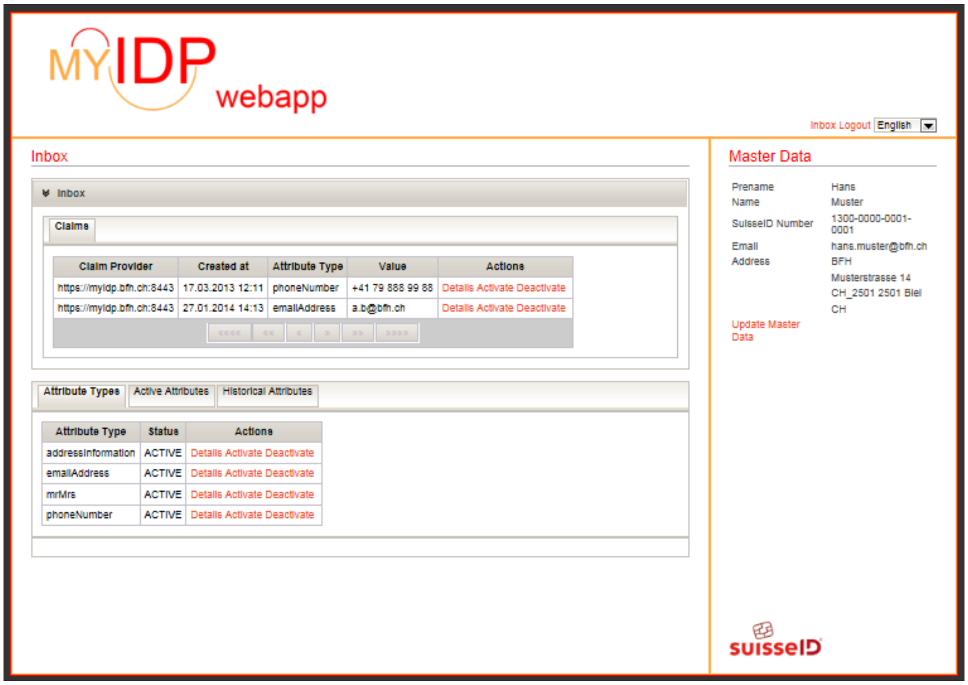


Figure 7. Screenshot myIdP WebApp - Inbox

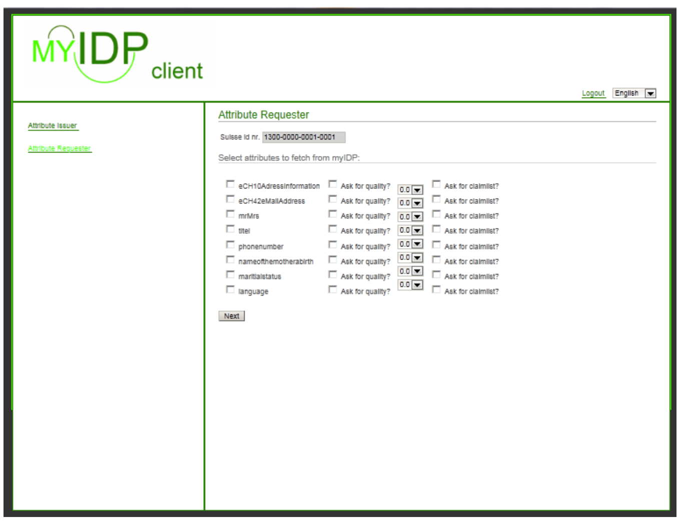


Figure 8. Screenshot myIdP Client - Attribute Requester

an unsolicited SAML response message to the myIdP Service. The myIdP service validates the message and if valid, stores it in the myIdP database. The user can now enter the myIdP WebApp to activate this attribute for further use.

Figure 8 shows the second role a service provider can incorporate. Before prompting the user for input, the service provider sends a request of a set of SAML attribute to the myIdP service. The user can now select and confirm the different values, instead of reentering them in the web application. In the myIdP client the normally hidden step is visualized. The user can select which attribute to request from myIdP. Additionally, a quality assurance value or the claim list activating the Claim Proxy mode of myIdP (see myIdP flavors in Section III-A) can be requested.

IV. Privacy

One important characteristic of myIdP (valid for both flavors) is the user-centric approach. The user is always aware which information is exchanged and has to confirm explicitly every single attribute that is sent out by myIdP.

myIdP implements multiple measures to ensure the privacy of the user:

First, every attribute issuer has to get a user consent before sending any attribute statement to myIdP. In the myIdP client, this is realized by actively requesting the user to check the check-box, that allows the application to send the information to myIdP (see Figure 6). Another option is to include the user consent prominent in the Terms of Usage of the application.

That the attribute issuers full fill these requirements is ensured by myIdP with a registration process for attribute issuers and a corresponding white list.

Secondly, in myIdP the incoming attributes are deactivated by default. The user has to confirm explicitly whether the attributes should be activated (for further use). At any time the user is free to delete attribute statements in the myIdP WebApp or to deactivate them.

Thirdly, the user is involved in every message exchange with an Attribute Requester and has to confirm all attribute values. In the claim proxy case, the disclosure of original attribute assertions needs to be approved as well. The attribute

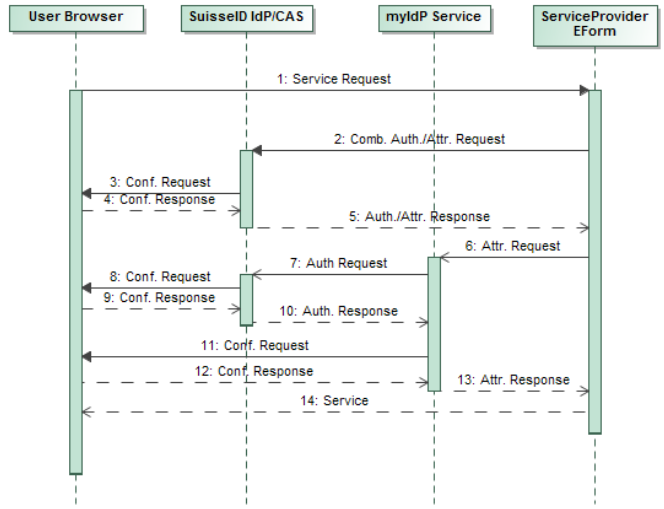


Figure 9. Sequence diagram "Get e-form"

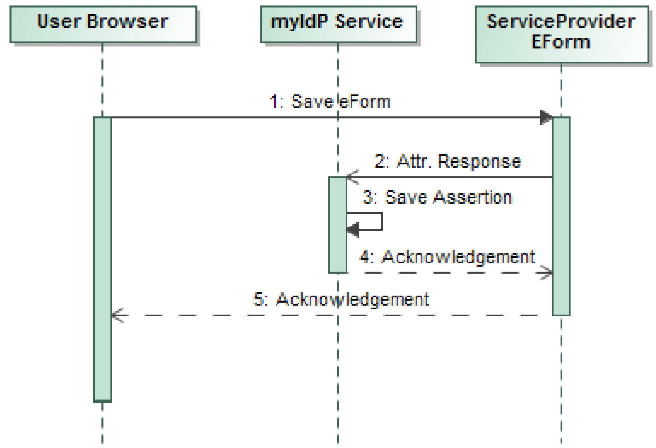


Figure 10. Sequence diagram "Save e-form"

assertions contain information about visited web sites and could be used to track the user and to create user profiles.

In any case, myIdP only sends attribute statements to Attribute Requesters if a valid authentication with a SuisseID in addition to the user consent exists. These procedures ensure that the user exposes only the intended data and the privacy is protected.

V. Application Scenario

A scenario of completing electronic forms (e-forms) validates our approach. E-forms are commonly used in the Swiss eGovernment. With the help of the SuisseID, the citizen can be identified securely and the attributes stored in the core SuisseID components, like name, birthday, place of birth or nationality, can be used to prefill the e-forms. As the number of attributes available in the core SuisseID is quite limited, we propose the usage of myIdP to provide additional values for the e-forms.

For our proof of concept, we chose the form "Proof of residence", which had an integration with the core SuisseID infrastructure already. In Figure 11, an extract of the French version of this form is displayed. The data filled from the core SuisseID infrastructure and myIdP are marked differently (pink - core SuisseID, yellow - myIdP). The data from the SuisseID cannot be overwritten by the user, as they represent certified attributes validated by a trusted authority. In contrast, the myIdP data can be updated. When all data is up-to-date, the user only has to enter one number, which is the number of copies wished-for (the field is marked with a red box), before sending the e-form to the administration.

In Figure 9, the interactions between the user, the e-form provider, the core SuisseID components and myIdP are depicted (only the main scenario is depicted, exceptions and error cases are omitted for the sake of readability):

1) Service Request: the user requests an e-form from the e-form provider (e.g., by clicking on a link).
2) Authentication with SuisseID: the e-form provider issues a combined authentication and attribute request to the SuisseID IdP/CAS. The following attributes are requested: name, first name and birthday.
3) Confirmation request: the user has to identify themself, by entering his secret key (PIN) and in a second step to confirm the SuisseID attributes.
4) Confirmation response: the user's decisions are sent back to the SuisseID IdP/CAS.
5) Authentication and Attribute response: the SuisseID IdP/CAS sends a combined authentication and attribute assertion back to the e-form provider.
6) myIdP attribute request: the e-form provider issues an attribute request to the myIdP service asking for the address and the email.
7) Authentication Request: the myIdP data are only accessible to the identified owners. That means, the myIdP Service forces a (second) SuisseID authentication.
8) Confirmation request: the user has to identify themself, by entering the secret key (PIN) and in a second step to confirm the disclosure of his identity.
9) Confirmation response: the user's decisions are sent back to the SuisseID IdP.
10) Authentication response: the SuisseID IdP sends an authentication assertion back to the myIdP Service.
11) Confirmation request: the user has to select the attribute values, in case several emails or addresses are stored in myIdP, and to confirm the selection.
12) Confirmation response: the user's decisions are sent back to myIdP.
13) Attribute response: myIdP sends an attribute assertion back to the e-form provider.
14) Service: the e-form is displayed to the user and contains the selected and confirmed values from the SuisseID IdP/CAS and myIdP.

The user now has to complete the e-form and to enter the number of copies desired. In case, the email or home address has changed, the data can be also manually corrected on the

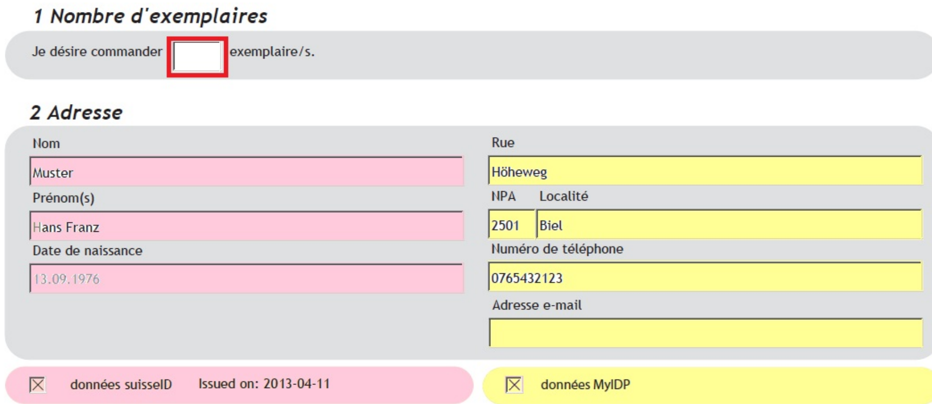


Figure 11. Prefilled e-form ”Proof of residence”

e-form (the data from the SuisseID IdP/CAS are read-only and cannot be changed). When the document is saved the governmental process of providing the requested documents is started. However, the confirmed data from the e-form are also transferred – as new attribute assertions (unsolicited message) containing validated information – to myIdP (see Figure 10) where it is stored in the myIdP database.

Looking closely at the interactions between the different actors involved in the application scenario (see Figure 9), it becomes obvious that the user has to authenticate themself twice with the SuisseID IdP: the first time to access the data from the SuisseID CAS and the second time to access the data from the myIdP service. This is quite inconvenient for the user and hardly acceptable in an eGovernment scenario. A possible solution is to enhance the myIdP Service further to support the proxying of authentication requests to a subsequent identity provider, as described in [22]. With this functionality, the user would be requested to authenticate only once; but the two attribute confirmation requests would still be necessary.

A crucial point of using myIdP in eGovernment applications and also in other domains is the selection and standardization of attributes. In our scenario, we could reuse attributes defined and published as Swiss standards, e.g., the eCH-0010 [23] for the address and eCH-0042 [24] for the email. Relying on these standards, the service provider (in our use case, the e-forms provider) can define a stable mapping between the field names in the e-forms or the web application and the attributes supported by the myIdP Service.

VI. Quality assessment and trustworthiness

As already mentioned in Section III-A, the myIdP Service is enabled to offer a quality assessment for the provided attributes. This is important, because myIdP does not provide certified information about the SuisseID owner, like a normal CAS. The source of a normal CAS is typically a register belonging to a public or private authority. Examples are the Health Professional Index or the Notary Index of Switzerland, available in [25].

myIdP provides personal information typically without an official authority, which could validate and certify this data. To ensure a good data quality and to increase the trustworthiness of the myIdP data, the user is not allowed to enter the data directly in myIdP. Only a registered attribute issuer can send assertions to myIdP. This ensures, that all information available in myIdP is validated at least once by a service provider acting as attribute issuer.

A service provider acting as attribute requester needs to know how reliable the myIdP data are. In a closed myIdP community, where all service providers incorporate both roles (attribute requester as well as attribute issuer) and have built up trust relationships, the myIdP data would be evenly trustworthy. In a more open environment, where many service providers interact with myIdP, this is different. The service providers need a reliable mean to ensure the trustworthiness of the myIdP information.

The myIdP offers a quality assessment based on an open model. When a service provider acts as an attribute requester, it can ask for the optional quality assessment by myIdP (a value between 0 and 1), provided together with the requested attributes. It can even insist of a certain quality and include a minimum quality the attribute must fulfill in the attribute request (visible in the screenshot in Figure 8). In this case, myIdP will only select attribute values which match or exceed the requested quality.

If a service provider is not willing to rely on the quality assessment of myIdP, he can choose the myIdP Proxy mode (see Section III-A). In this myIdP flavor, the attribute requester gets all stored attribute assertions from the myIdP database belonging to the return attribute value. They can now perform their own quality assessment.

The quality assessment is a statement about the potential correctness of an attribute value. We identified three factors the quality assessment in myIdP should be based on:

1) Freshness of information
2) Quality of attribute issuer
3) Recurrence of information

A. Freshness of information

The freshness f can be calculated from the age a of an attribute assertion. This is the time between when the attribute assertion was issued and the time when a service provider requests this information. The fresher the attribute assertion, the better the quality. The quality of an attribute decreases gradually. In [26], the Formula (1) was elaborated and tested. It calculates the freshness on the basis of a normalized age value (like the quality a value between 0 and 1). The normalization has to be determined in dependency of the attribute. The average validity of attributes can be quite different; some, like eye color or gender, do almost not change during lifetime, others, like address, do in certain periods. To determine this average validity, demographic information could be used.

$$
\begin{aligned}
f &= 0.5 + \sqrt{1 - 2a^2} * 0.5 \quad \text{if } (0 \leq a \leq 0.5) \\
f &= 0.5 - \sqrt{1 - 4 * (a-1)^2} * 0.5 \quad \text{if } (0.5 < a \leq 1)
\end{aligned} \tag{1}
$$

B. Quality of attribute issuer

We propose to classify the attribute issuers according the the STORK Attribute Quality Authentication Assurance (AQAA) scheme [27], which is an extension of the STORK Quality Authentication Assurance (QAA) scheme published in [28]. The STORK QAA model permits quality levels to be assigned to various eID solutions, based on some of their main characteristics. The STORK AQAA model extends this model to be applied to attribute providers providing no directly related information to an eID solution.

The myIdP attribute issuers can be considered as attribute providers and therefore be classified into the four STORK QAA Levels (see Table I).

TABLE I. STORK QAA LEVELS [28]

STORK QAA level	Description
1	No or minimal assurance
2	Low assurance
3	Substantial assurance
4	High assurance

The AQAA level of the attribute issuers influences the quality of an attribute, like shown in Figure 12. In combination with the freshness of attribute (see Formula (1)), a low level of AQAA results in a displacement of the curve and therefor in a general decrease of quality.

The quality q of a single assertion can be calculated using the Formula (2), whereby coefficient k_{AQAA} indicates the decrease of quality assigned to the reached AQAA level of the attribute issuer.

$$q = \max\{f - k_{AQAA}, 0\} \tag{2}$$

C. Recurrence

An increasing number n of assertions containing the same value for an attribute should increase the quality of this value. We propose a formula (3) that shows a logarithmic behavior to calculate the recurrence r_{set} of a set of assertions. The rise coefficient k_{rise} is responsible to determine how steep the increase of quality should be at the beginning; it is normally

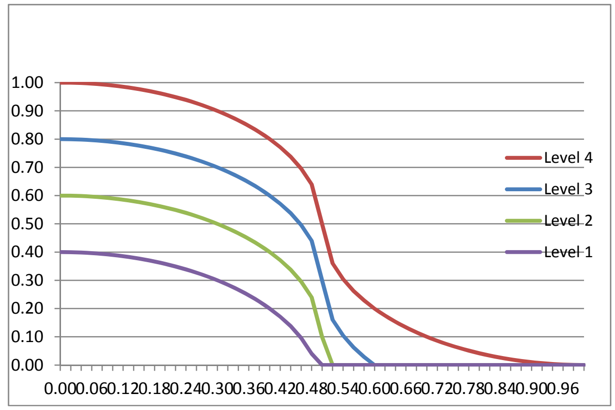


Figure 12. Freshness combined with AQAA level of attribute issuers

a value between 1 and 10. Like the influence of the age of the freshness, the rise coefficient can be different for each attribute.

$$r_{set} = \min\{(\log(n) + k_{rise})/(k_{rise} + 1) * 0.5, 1\} \quad \text{if } (0 \leq n) \tag{3}$$

D. Assertion sets

The quality of an assertion set can be calculated using different formulas. The simplest case would be to take only into account the recurrence and use Formula (3).

In Formula (4), the best freshness value is combined with size of the set (recurrence). The AQAA level of attribute issuers is not taken into account.

$$q_{set} = \min\{\max_i\{f_i\} + r_{set}, 1\} \tag{4}$$

Formula (5) is quite similar to Formula (4), but it uses the freshness in combination with the AQAA level.

$$q_{set} = \min\{\max_i\{q_i\} + r_{set}, 1\} \tag{5}$$

Both formulas tend to overrate the recurrence of the information and are prone to fraud scenarios, where many assertions from a low ranked attribute provider can significantly increase the quality.

$$q_{set} = \min\{\max_i\{q_i\} + r_{set}, \alpha_{MaxAQAA+1}\} \tag{6}$$

The third formula (6) reduces this risk of fraud by limiting the quality of the set of assertions to the maximum value allowed by the next highest AQAA level ($\alpha_{MaxAQAA+1}$). That means, when the set contains hundreds of assertions from a low-level attribute provider, the quality cannot reach a better value than the highest possible quality of an assertion from an attribute provider on the next higher level.

All three formulas were validated in several scenarios. Still, an assessment of a live-running scenario in a set-up myIdP community is lacking. On the basis on real-world data, also other approaches to calculate the quality assessment, e.g., based on subjective logic or Bayesian networks would be possible.

VII. Conclusion

myIdP is an extension to the SuisseID infrastructure. It proposes a Claim Assertion Service (SAML attribute authority), which handles personal data used and validated beforehand in other internet transactions. The concept is extensible to other eID solutions and can also be integrated in the STORK European eID Interoperability Platform. In a next step, the possibility to use myIdP as OpenID attribute provider will be investigated. Also the combination with a WebID seems feasible.

The myIdP concept was validated with a prototypical implementation following the proposed architecture. The initial implementation on the basis of the SuisseID SDK [29] quickly showed some limitations. Especially the use of an flexible attribute set or structured attributes, like address, were only partly supported. This was also due to the SuisseID SDK, which was designed for a fixed attribute set. These limitations were addressed in a subsequent bachelor thesis [30]

As proof of concept, the prototype was integrated in an eGovernment scenario of prefilling an e-form in order to obtain a proof of residence. The integration of more e-forms is planned. As precondition the set of myIdP attributes has to be extended to have a standardized basis for the information exchange.

The promoting of the myIdP service showed that many applications are willing to act as Attribute Requester and to use the personal attributes available in myIdP. The functionality to act as Claim Provider and to provide validated information to myIdP and to confirm the reuse is often seen as burden. However, both roles equally have to be provided to create a network of validated personal attributes.

As soon as more service providers will use myIdP in a life scenario and provide regularly attribute claims, the model to calculate the assurance level (quality) can be validated on a real data basis and be improved further.

To strengthen the user-centric approach even more and to protect the private attribute, the central storage of claims in the myIdP database could be changed towards a pseudo-local approach that lets the user choose where to store the data: on a personal device or on a central place. The storage of SAML assertions on the user's device would also enable the usage of myIdP - in addition to the normal online scenario - in environments with limited or no connectivity.

Acknowledgment

Thanks to all students from the Bern University of Applied Sciences who helped with their student projects and bachelor theses [31][32][26] to realize this project.

References

[1] A. Laube and S. Hauser, "myIdP-The Personal Attribute Hub," in Proceedings of the Fifth International Conferences on Advanced Service Computing (SERVICE COMPUTATION 2013), Valencia, Spain, 2013, pp. 1–5.

[2] Arbeitsgruppe Spezifikation des Trägerschaftsverein SuisseID, "eCH0113 SuisseID Specification, Version 1.5," November 30, 2011.

[3] Quo Vadis - SuisseID - Website. [Online]. Available: http://www.quovadisglobal.ch/de/Zertifikate/SuisseID.aspx [retrieved: June, 2014]

[4] Swiss Post - SuisseID - Website. [Online]. Available: http://postsuisseid.ch/en [retrieved: June, 2014]

[5] Arbeitsgruppe SuisseID c/o Staatssekretariat für Wirtschaft SECO, "Claim Assertion Service (CAS), Technical Specification, Version 0.99.07," January 13, 2011.

[6] N. Ragouzis et al., "Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS Committee Draft," March 2008. [Online]. Available: http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

[7] J. Hodges, R. Philpott, and E. M. (Ed.), "Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: https://www.oasis-open.org/committees/download.php/21111/saml-glossary-2.0-os.html

[8] M. Margraf, "The new German ID card," February 2011. [Online]. Available: http://www.personalausweisportal.de/SharedDocs/Downloads /EN/Paper_new_German_ID-card.pdf

[9] The official beID website. [Online]. Available: http://eid.belgium.be/en/ [retrieved: June, 2014]

[10] Austrian Citiziens Card - Official Website. [Online]. Available: http://www.buergerkarte.at/en/index.html [retrieved: June, 2014]

[11] STORK - Project Website. [Online]. Available: https://www.eid-stork.eu [retrieved: June, 2014]

[12] STORK 2.0 - Project Website. [Online]. Available: www.eid-stork2.eu [retrieved: June, 2014]

[13] specs@openid.net, "OpenID Authentication 2.0 - Final," December 2007. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html

[14] Clavid - Official Website. [Online]. Available: clavid.ch [retrieved: June, 2014]

[15] Cloudid.de - OpenIDentity Provider - Website. [Online]. Available: cloudid.de [retrieved: June, 2014]

[16] Google, "Federated Login for Google Account Users," June 2012, accessed June 2014. [Online]. Available: https://developers.google.com/accounts/docs/OpenID

[17] D. Hardt, J. Bufu, and J. Hoyt, "OpenID Attribute Exchange 1.0 - Final," December 2007. [Online]. Available: http://openid.net/specs/openid-attribute-exchange-1_0.html

[18] Open AXN group. Street Identity - Website. [Online]. Available: https://sites.google.com/site/streetidentitylmnop/ [retrieved: June, 2014]

[19] H. Story and S. Corlosquet (eds.), "WebID 1.0. Web Identification and Discovery. W3C Editor's Draft." January 2013. [Online]. Available: http://www.w3.org/2005/Incubator/webid/spec/

[20] D. Brickley and L. Miller, "FOAF Vocabulary Specification 0.98," August 2010. [Online]. Available: http://xmlns.com/foaf/spec/

[21] S. Cantor, J. Moreh, R. Philpott, and E. M. (Ed.), "Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[22] S. Cantor, J. Kemp, R. Philpott, and E. M. (Ed.), "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[23] Verein eCH, “eCH-0010: Datenstandard Postadresse für natürliche Personen, Firmen, Organisationen und Behörden,” October 2011. [Online]. Available: http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0010&documentVersion=5.00

[24] ——, “eCH-0042: Vorgehen zur Identifizierung von eGovernment-relevanten Geschäftsinhalten,” June 2005. [Online]. Available: http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0042&documentVersion=1.00

[25] Notary Register of Switzerland - Website. [Online]. Available: https://reg.sdms.ch/eRoleRegister/SID-7dcb2c010099bd3ff8143569fdb840c3/Notary/Index [retrieved: June, 2014]

[26] A. Keller, “Qualitätsmodell im Kontext von myIdP. CASE Arbeit.” Master’s thesis, Bern University of Applied Sciences (BFH), June 2012. [Online]. Available: http://www.myidp.ch/acms/fileadmin/documents/case_Qualitaetsmodell_v1.0.pdf

[27] H. Graux, “D.3.2 - QAA status report,” Apr 2013.

[28] B. Hulsebosch, G. Lenzini, and H. Eertink, “D2.3 - Quality authenticator scheme,” Mar 2009.

[29] SuisseID SDK - website. [Online]. Available: https://www.e-service.admin.ch/wiki/display/suisseid/Home [retrieved: June, 2014]

[30] E. Jeannerat, C. Saner, and D. Schaeffer, “Bachelorthesis: SuisseID V2.0 SDKs,” Master’s thesis, Bern University of Applied Sciences (BFH), June 2013.

[31] R. Imwinkelried and D. Ehrler, “Bachelorthesis: Specification myIdP,” Master’s thesis, Bern University of Applied Sciences (BFH), January 2012.

[32] R. Bühlmann and M. Jeker, “Bachelorthesis: Specification myIdP Extensions,” Master’s thesis, Bern University of Applied Sciences (BFH), June 2012.

A Robust Client-Driven Distributed Service Localisation Architecture

Luke Collins, Ming-Xue Wang, and Claus Pahl
School of Computing
Dublin City University
Dublin 9, Ireland
Email: luke.collins4@mail.dcu.ie, mwang@computing.dcu.ie, claus.pahl@dcu.ie

***Abstract*—The fundamental purpose of service-oriented computing is the ability to quickly provide software resources to global users. The main aim of service localisation is to provide a method for facilitating the internationalisation and localisation of software services by allowing them to be adapted to different locales. We address lingual localisation by providing a service interface translation using the latest web services technology to adapt services to different languages and currency conversion as an example of regulatory localisation by using real-time data provided by the European Central Bank. Units and Regulatory Localisations are performed by a conversion mapping, which we have generated for a subset of locales. The aim is to investigate a standardised view on the localisation of services by using runtime and middleware services to deploy a localisation implementation. We apply traditional software localisation ideas to service interfaces. The architecture here is client-centric, allowing the localisation to be controlled and managed by the client, ultimately providing more personalisation and trust. It also addresses robustness concerns by enabling a fault-tolerant architecture for third-party service localisation in a distributed setting. Our contribution is a localisation platform consisting of a conceptual model classifying localisation concerns and the definition of a number of specific platform services.**

Keywords - Service Localisation; Service-oriented Computing; Service-oriented Architecture.

I. INTRODUCTION

Distributed web services can provide business and private consumers with computing abilities, which may not be feasible for them to develop in-house. These web services are currently in high demand in the context of cloud computing [1]–[5]. However, the area of services computing introduces new issues, for example, in areas like Europe, where there is a wide range of languages spoken, services are very often only developed for single language and are only supported for that single language. Equally, adapting services to different regulatory environments with different legal systems, taxation and units in place is equally challenging. Often it is the case that companies do not have the resources or capability to develop multilingual products. Localisation encapsulates a large number of issues, which need to be addressed in this context. These include, but are not limited to:

- Language Translation - conversion of services based on language. e.g., $English \rightarrow French$.
- Regulatory Compliance Constraints - conversion of services based on information such as taxation and other regulatory constraints.
- Currency Conversion - conversion of services based on currency, e.g., $Euro \rightarrow Dollar$.
- Units Conversion - based on standard units measurements, e.g., $Metric \rightarrow Imperial$.

Further concerns such as standardised vocabularies and conventions could be added.

Localisation is typically performed on textual content (i.e., strings) and refers to either languages only or physical location. However, the purpose of this work is to develop a method of localising services by implementing a 'mediator' type service, which interacts between the Application Programming Interfaces (APIs) of the service provider and the requester. This mediator largely automates the service interface localisation process. We are going to focus on a number of locale dimensions such as language, taxation, currency and units. An example of a request that requires localisation can be seen in Figure 1, which illustrates an example of a financial service provided to a range of locales (locations or regions requiring equal conversions).

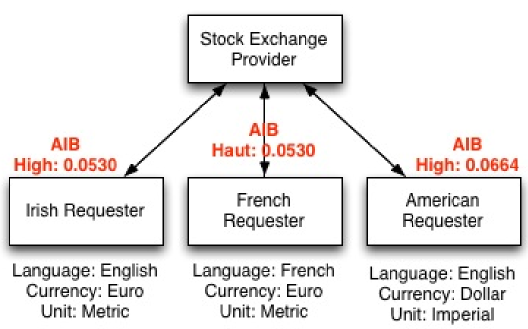


Figure 1: Overview of Requests Requiring Localisation

We look at service-level language translation techniques to localise services (including APIs) to different languages as part of a lingual translation idea. Regulatory translation, which includes currency, units and taxation as legal governance and compliance rules, will be provided by standards-based mappings. Regulatory translation is important for applications to comply with varying regional laws and regulations.

The objectives of service localisation include primarily the introduction of service-centric localisation techniques. A specific need is to make localisation techniques available at runtime for dynamic localisation, which is required for currencies and other variable aspects. A localisation mediator takes care of this task. Thus, Service Localisation (SL) provides a mechanism for converting and adapting various digital resources and services to the locale of the requester. A greater end-to-end personalisation of service offerings is an aim. A Localisation Provider act as an intermediary between the service provider and the requester. In our proposed platform, this is supported by a mediation service. We will provide a novel architecture where in addition to the new concept of service interface localisation, this can even be controlled by the user at the client-side.

By generating a common platform solution for these localisation issues, we allow the ability to dynamically localise Web services to be made with little effort. Our key contributions are:

- Software Localisation at Service Level - the main concern is a standardised mapping within a potentially heterogeneous environment.
- Adaptation and Integration - the main concern is the maintenance of service quality after it has been localised through adaptation.
- Client-side Control - the main concern is a robust, fault-tolerant coordination solution that allows localisation to be managed client-side.

The novelty of the proposed solution lies in filling a gap between service adaptation techniques (largely ignoring the regulatory and lingual aspects) and existing service internationalisation, which looks into localisation concerns, but only to a basic extent covering data formats and unit and currency conversions. An important aspect of this investigation is a robust coordination platform that not only allows service consumers to define and manage the localisation behaviour, this platform also needs to be able to address the challenges of services provided across a distributed setting with failure and non-applicability of localisation policies as a consequence. We aim to show through a concrete example an appropriate use of service localisation. The example also attempts to illustrate various benefits and use cases. We also discuss motivating factors behind using a localisation technique.

In the next section, we discuss the motivation behind developing a Service Localisation implementation. Section III defines a platform architecture for Service Localisation. In Section IV, we introduce aspect-specific localisation techniques, which we investigated and implemented. In Section V, we investigate the coordination solution for the client-side management of the localisation settings. Section VI introduces the implementation and evaluates our solution to the Service Localisation problem. Section VII contains the related work discussion. In Section VIII, future directions and possible extended infrastructures are explored.

II. State-of-the-Art and Motivation

Our main focus is a platform for service localisation, which makes a shift from the typical ”one size fits all” scenario towards a more end-to-end personalised service scenario. Currently, services computing suffers from localisation and adaptability issues for multiple users in different regions. These issues could be overcome if a multi-lingual and multi-regional solution was developed [6], [7].

A. Motivating Scenarios

The different localisation issues of a service can be illustrated. The scenarios described below are used to illustrate benefits to service localisation:

- End-User Services: Some software-as-a-Service providers only support one region with one specific language. There is a possibility to perform localisation both statically (compile-time) and dynamically (runtime), which typically involves localising service values and interacting messages.
- Business Services: Various business centric applications including applications for documentation and analysis could be localised to support various legal and regional locales. Business services typically require more customisation than end-user consumers.
- Public Sector Services: As governments outsource their computing infrastructure to external providers, it is becoming more important for the providers to supply solutions, which take into account various regulatory governance aspects such as currency and taxation and also lingual localisation.

Another scenario, which provides a detailed view of the benefits of service localisation, could be a service provider, used to manage company accounts for its customers. This could be a company that has offices in different global locations and would like to provide localisation based on customer region and localisation for its individual offices.

- Regulatory: Conversion of data between standards and their variants, e.g., based on different units of measurement $Metric \rightarrow Imperial$.
- Currency: Conversion of between currencies, e.g., $Euro \rightarrow Dollar$.
- Lingual: Translation of service related data between languages. This could include free text, but also specific vocabularies based on business product and process standards such as GS1 or EANCOM.
- Taxation: Different customers have different taxation requirements, e.g., VAT rates. Localisation of accounts software can take this into account for each locale.

B. Use Cases and Requirements

In order to demonstrate the need for localisation of Web services, we chose to demonstrate the issue using a concrete case of an environment, which utilises service-level access to a stock exchange interface. Imagine an Irish user who wishes to access data from the New York Stock Exchange, which is provided in an English format with the currency in dollars. A user in France may also wish to access data from the New York Stock Exchange using a French interface where local regulations require French to be used for data and/or service operations. Therefore, there must be a mechanism to convert the currency to Euro or to another currency that the requester specifies. There must also be a mechanism to convert the language to that of the requester.

At application level, two sample calls of a stock exchange data retrieval service for the two different locales (IE-locale with English as the language and EUR as the currency and FR-locale with French as the language and EUR as the currency) retrieve the average stock price for a particular sector - in this case the financial sector as follows:

- $Retrieve(20/08/2012, Financial) \rightarrow 30.50\,EUR$
- $Récupérer(20.08.2012, Financier) \rightarrow 30,50\,EUR$

In the US-locale with English as the language and USD as the currency, the same API call could be the following:

- $Retrieve(08/20/2012, Financial) \rightarrow 38.20\,USD$

The following elements in this case are localisable:

- Date: in order to preserve regulatory governance, the date format requires to be changed depending on the requester locale.
- Language: names of functions from the API are translated between languages.
- Currency: values are converted as normal and this would apply to any other units.

This list can vary depending on the environment where different regulatory constraints might apply. In general, it can be expected that there is always a linguistic element to the localisation of any product, but elements may also include taxation and units of measurement. If it was the case that the requesters were trying to access weather forecasts for their own region and formatted in their own locale, then it would be necessary to utilise a conversion for units of measurement:

- $Prévision(20.08.2012) \rightarrow 30°Celsius$
- $Forecast(20/08/2012) \rightarrow 15°Celsius$

In the US-locale with English and imperial units, the same API call could be $Forecast(08/20/2012) \rightarrow 87°Fahrenheit$.

Note that this scenario takes real-world services into account. Companies like xignite (provides stock market analysis services) and Deutsche Boerse (provides access to Frankfurt's stock market information services) have published and implemented services underlying the examples here.

III. Localisation Framework

Localisation of service interfaces requires a framework to facilitate various localisation methods. These various methods, implemented as services in our proposed localisation architecture, are used to facilitate the localisation of localisable elements or artefacts. This paper focuses on the dynamic localisation of service-level interface descriptions.

A. Information Architecture

With every service there are various elements that may be localised. These elements include:

- Service specifications/descriptions (APIs)
- Models (structural/behavioural)
- Documentation (for human consumption)
- Messages exchanged between services

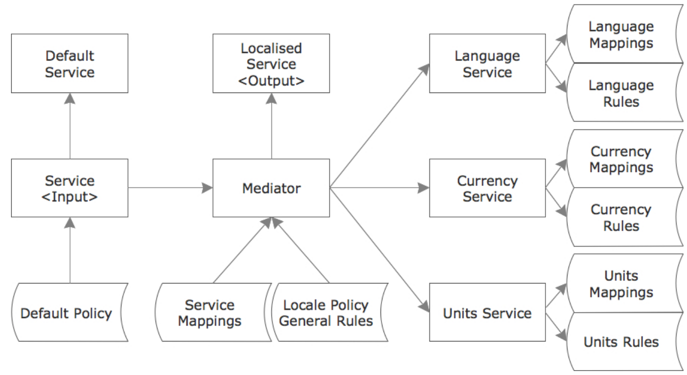


Figure 2: Conceptual Architecture of the Localisation Platform.

Services are normally written to be independent of locales, however localisation is often needed to further personalise or adapt a service to specific contexts. A localisation platform should be based on attributes that vary from locale to locale, like time or date format or the language.

A service localisation platform requires a number of elements. These elements can be pre-translated fragments in static form or can be dynamic translation systems. Figure 2 aims to demonstrate the concept of a policy and mappings based system architecture, which can be scaled when additional processes are attached to the mediation process. In the platform architecture, user-specific locale policies are applied to service endpoints. For example, in a WSDL file we may localise messages and operation names. Rules for each type of translation would be stored in a rules database (General Rules Repository) [8]. Similarly, mappings between common translations would be stored in a mappings database (Translation Memory). Note, that we will discuss the distribution and management of services between client and provider in Section V. The central mediator component from Figure 2 requires localisation adaptation to be facilitated dynamically. In Section V, a coordination platform for the dynamic interaction between service clients and providers is defined where the clients can control the adaptation through the mediator.

B. Systems Architecture

A mediator operates between users (with different locales) and several service providers (with different locales) by providing core localisation services, such as currency conversion and language translation. The architecture supports the following:

- Static Mappings: these could be the mapping of one language to another or one unit to another, pre-translated in translation memories.
- Dynamic Localisation: when translation mappings are not stored, dynamic localisation is required in order to obtain a correct translation and store the mapping.
- Policy Configuration: in order to configure the various locale policies, we must generate particular translation rules, supported by a logical reasoning component.
- Negotiation: this is the exchange of locale policies through the form of XML and SOAP from a web

services point of view.

- Localisation of Services: the mappings between the remote service and the localised service description must be stored in a mappings database (Translation Memory) so the localised service has a direct relationship with the remote service.

The workflow of the mediator process is $Negotiation \rightarrow PolicyConfiguration \rightarrow Localisation \rightarrow Execution$.

Some examples shall illustrate the functionality of the platform. Table I defines two different locales in the format of XML profiles. A mismatch between the requester locale and the provider locale needs to be bridged by the mediator localisation service. The language as a lingual aspect and country, currency and unit codes are regulatory concerns.

TABLE I: Sample Environment Setup

```
<SLContext>
   <Locales>

      <RequesterLocale>
        <LanguageCode>efr </LanguageCode>
        <CountryCode>FR</CountryCode>
        <CurrencyCode>EUR</CurrencyCode>
        <UnitCode>M</UnitCode>
      </RequesterLocale>

      <ProviderLocale>
         <LanguageCode>en </LanguageCode>
         <CountryCode>IE </CountryCode>
         <CurrencyCode>EUR</CurrencyCode>
         <UnitCode>M</UnitCode>
      </ProviderLocale>

   </Locales>
</SLContext>
```

The locale definitions decide how a given service API (in the Web service description language WSDL) is localised. Results from a sample execution of the localisation service (the mediator) is displayed in Tables II and III based on the XML locale definitions of the environment in Table I. Table II shows excerpts from an original WSDL file. Table III shows the localised WSDL after the application of lingual localisation in this case (translation from English (IE locale) into French (FR locale) – for simplicity of the example, we have focused on this single aspect only), compliant with the two locale definitions from the first listing.

TABLE II: Sample Input - Provider Locale

```
<wsdl:message name="quoteResponse">
  <wsdl:part name="parameters"
             element="quoteResponse"/>
</wsdl:message>
<wsdl:message name="quoteRequest">
  <wsdl:part name="parameters"
             element="quote"/>
</wsdl:message>
<wsdl:portType name="Quote">
  <wsdl:operation name="getQuote">
    <wsdl:input name="quoteRequest"
                message="quoteRequest"/>
    <wsdl:output name="quoteResponse"
                message="quoteResponse"/>
  </wsdl:operation>
</wsdl:portType>
```

TABLE III: Sample Output - Localised WSDL

```
<wsdl:message name="quoteReponse">
  <wsdl:part name="parameters"
             element="quoteReponse"/>
</wsdl:message>
<wsdl:message name="citerDemande">
  <wsdl:part name="parameters"
             element="citer"/>
</wsdl:message>
<wsdl:portType name="Citer">
  <wsdl:operation name="getCiter">
    <wsdl:input name="citerDemande"
                message="citerRequest"/>
    <wsdl:output name="citerDemande"
                message="citerDemande"/>
  </wsdl:operation>
</wsdl:portType>
```

IV. Localisation Rules and Services

We have outlined the core platform architecture in the previous section with the central services. In order to provide the localisation platform services, we need to realise a number of localisation services to enable a modular service localisation platform. Their interaction is summarised in Figure 3. Details of underlying concepts of their operation are explained now. We will discuss the topology, i.e., where the individual services are provides and who manages them, behind this interaction specification in the following Section V.

A. *Rule-based Locale Definition and Conversion*

At the core of our service localisation platform is a language to specify the localisation policy rules in relation to localisations. In most cases, languages like WSDL and other XML languages provide information regarding the services that are provided via an API. However, in order to encapsulate localisation information, there is a necessity to provide a language that will contain details in relation to the locales of the requester and the provider. For the purpose of our localisation platform, we use a policy language based on the Semantic Web Rule Language SWRL, which is based on the propositional calculus.

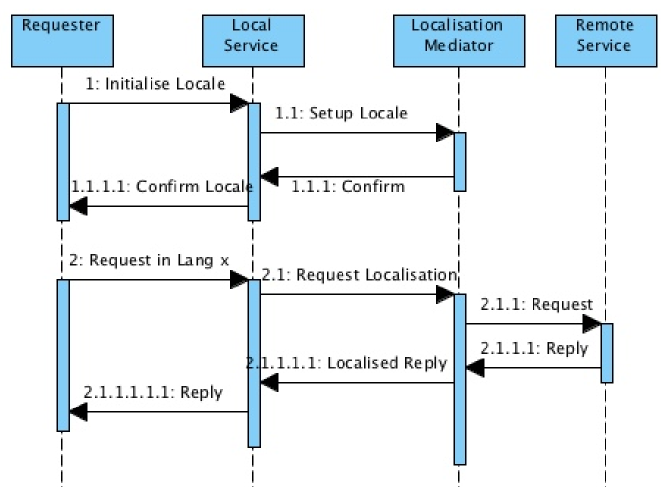


Figure 3: A UML Sequence Diagram of the Platform.

A localisation layer encapsulates the various forms of translations. It describes the relationships between localisable elements. For example, it contains the details of items that can be translated. For our localisation model these are documentation and descriptions, but also API messages and operations. The rule language is used to define localisation policies of two types: firstly, locale definitions and, secondly, conversion (translation) rules. We motivate the rule set through examples.

Firstly, there are a number of locale definition rules provided, like *Loc* or *hasCur*, by which locales for specific regions are described. A locale can also be described by other rules such as *hasTax*, *hasLang* and *hasUnit*. Examples of three region's locales - IE, US, and FR - are:

$$IELoc(?l) \leftarrow Loc(?l) \wedge$$
$$hasLang(?l, ?z) \wedge hasCur(?l, ?c) \wedge hasUnit(?l, ?u) \wedge$$
$$?z = en \wedge ?c = EUR \wedge ?u = metric$$

$$USLoc(?l) \leftarrow Loc(?l) \wedge$$
$$hasLang(?l, ?z) \wedge hasCur(?l, ?c) \wedge hasUnit(?l, ?u) \wedge$$
$$?z = en \wedge ?c = USD \wedge ?u = imperial$$

$$FRLoc(?l) \leftarrow Loc(?l) \wedge$$
$$hasLang(?l, ?z) \wedge hasCur(?l, ?c) \wedge hasUnit(?l, ?u) \wedge$$
$$:?z = fr \wedge ?c = EUR \wedge ?u = metric$$

The benefit of a formal framework for the rules is that other rules can be inferred by from partial information. For example, if we knew that a locale had USD as its currency we may be able to infer its country from it:

$$?c = USD \rightarrow ?l = USLocale.$$

These inferred rules do not apply in general - this may not work if we know a currency is Euro in which case it could be one of many locales in Europe. However, the purpose of these rules could be to determine inconsistencies. Preconditions can clarify the remit of these rules.

Secondly, a generalised conversion between locales, e.g., $Locale\ A \rightarrow Locale\ B$, is given by the following general conversion rule:

$$IELoc2USLoc(?l1, ?l2) \leftarrow$$
$$hasLang(?l1, ?z1) \wedge hasLang(?l2, ?z2) \wedge$$
$$hasCur(?l1, ?c1) \wedge hasCur(?l2, ?c2) \wedge$$
$$hasUnit(?l1, ?u1) \wedge hasUnit(?l2, ?u2) \wedge$$
$$?z2 = convertLang(en, en, ?z1) \wedge$$
$$?c2 = convertCur(EUR, USD, ?c1) \wedge$$
$$?u2 = convertCur(metric, imperial, ?u1)$$

$$IELoc2FRLoc(?l1, ?l2) \leftarrow$$
$$hasLang(?l1, ?z1) \wedge hasLang(?l2, ?z2) \wedge$$
$$hasCur(?l1, ?c1) \wedge hasCur(?l2, ?c2) \wedge$$
$$hasUnit(?l1, ?u1) \wedge hasUnit(?l2, ?u2) \wedge$$
$$?z2 = convertLang(en, fr, ?z1) \wedge$$
$$?c2 = convertCur(EUR, USD, ?c1) \wedge$$
$$?u2 = convertCur(metric, metric, ?u1)$$

Depending on client and provider locale, any combination of mappings/translations can be generated by the core rules.

B. Localisation Mediator

Based on these locale policy definitions and conversion rules, a number of services operate. In order to provide a transparent localisation system, a central component acts as a mediator, as visualised in Figure 4, which in turn uses individual services for: Lingual Conversion, Currency Conversion, Regulatory Governance, Units Conversion, and WSDL Parsing & Generation. Within this mediator architecture, the mediator methods call the other localisation services of the platform.

During execution of the localisation platform, an XML filewith the ploicy definition is first passed to the mediator (we will look at the underlying coordination mechanism for this in Section V). The Mediator Service then sets up a localisation environment using the locale details provided in *LocaleConfig.xml*, the component performs this via the use of the respective interfaces. Once the locale is set up, the service Web Service Description Language (WSDL) file is parsed and various elements are localised resulting in a localised WSDL file that can be used to access localised operation mappings. This component is the work horse of the platform and can be extended with the introduction of other localisation classes, i.e., the architecture is modular.

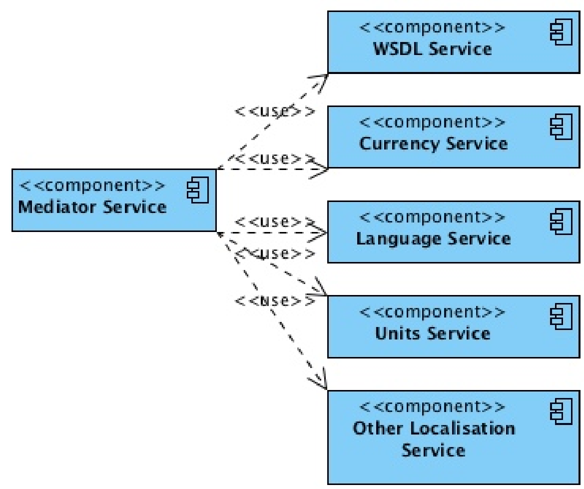


Figure 4: A Component Diagram Displaying Extensibility.

Linguistic artefacts are one of the most widely localised elements of software today. We propose machine translation (MT) to achieve automation. While further research into a tailored MT solution is required to specifically address limited textual context and controlled vocabularies for APIs, language translation within the proposed platform is provided by the Google Translate API. In the interest of performance, our platform tries to make as few API calls to Google as possible. Instead it stores translations of popular words and glossaries within a local language mapping database (a Translation Memory) for later retrieval. A local machine translation system may also reduce this latency, as it would no longer have to depend on TCP/IP performance. The conversion rule for language translation is given by:

$$IELoc2FRLoc(?l1, ?l2) \leftarrow$$
$$hasLang(?l1, ?z1) \wedge \quad hasLang(?l2, ?z2) \wedge$$

$?z2 = convertLang(en, fr, ?z1)$

Regulatory localisation through adaptation to other regulatory standards is based on localising regulatory concerns. These concerns include, but are not limited to the following: Taxation, Currency, and Units of Measurement. We have chosen to localise a subset of these concerns. For the purpose of units localisation, we developed an interface to a repository of unit conversion formulae. These formulae provide conversions between the metric and imperial units of measure. The conversion rule for units is given by:

$$IELoc2USLoc(?l1, ?l2) \leftarrow$$
$$hasUnit(?l1, ?u1) \wedge hasUnit(?l2, ?u2)$$
$$\wedge\, ?c2 = convertUnits(metric, imperial, ?u1)$$

Due to a large number of currencies used globally, we propose a separate service to deal with currency conversion. For the purpose of currency localisation, we use exchange rates from the European Central Bank. This is in our case supported by a MySQL database. Currencies are manipulated based on their rate compared to Euro as the base currency. The conversion rule for currency is given by:

$$IELoc2USLoc(?l1, ?l2) \leftarrow$$
$$hasCur(?l1, ?c1) \wedge hasCur(?l2, ?c2)$$
$$\wedge\, ?c2 = convertCur(EUR, USD, ?c1)$$

In order to parse the input in the form of WSDL files, a WDSL service is used. This contains the methods required to manipulate both incoming WSDL files of the service provider and has the ability to generate a localised WSDL file. The service can be considered as an I/O Manager. XLIFF is an XML standard for translation that proved useful when it comes to the localisation of WSDL file.

V. LOCALISTION - COORDINATION AND INSTRUMENTATION

Figures 2 and 3 define the architecture in abstract terms. A key feature of our solution is the possibility for clients to define the localisation constraints and policies and to manage the localisation themselves to achieve a higher degree of dynamic personalisation. In Figure 3, the Local Service is a client-side localised facade to the actual basic service as provided server-side. The mediator handles the required location as discussed in the previous Sections III and IV. In order to manage the client-side definition and enforcement of localisation policies, a coordination framework is necessary, which is described in this section. The adaptation needs to be client-side controlled as there there specific localisation context is defined. It also relieves the provider from the customisation activities.

For both the mediator and the server side, we assume BPEL process engines to manage the processes, like the mediator process $Negotiation \rightarrow PolicyConfiguration \rightarrow Localisation \rightarrow Execution$ that we introduced earlier. These generic processes and their constituent services need to be adapted to the needs specific localisation policies.

A coordination framework for localisation with protocols as the implementation of the localisation makes process consumers and providers contribute together to localisation to ensure that defined policies are enforced. For a localised service requested by a process consumer, there are a number of activities including those from subprocesses within a process that will participate in the coordinated execution (a kind of transaction is required). The WS-Coordination specifications are designed for transactions of distributed Web services rather than transactions of application processes. Adaptive processes for handling processes transactions lack coordination mechanisms for our case to guarantee all participants working together in a unified manner. The coordination framework we implemented for this localisation context addresses these limitations. It includes suitable protocols for the participants for any application process.

The localisation framework uses a mix of local localisation services (e.g., unit conversion), external services (currency conversions) and hybrid techniques (e.g., for translation). This mix of widely distributed services makes a coordination solution necessary that takes failure into account. Services might become unavailable. Localisation policies might not be applicable as a consequence. A solution that allows the applicability of policies to be check prior to localisation execution or post-execution are therefore required [9].

We first introduce a coordination model that focuses on message exchange or coordination contexts between clients and mediators that act as coordinators. A coordination protocol for localisation policy enforcement in service transactions is also defined. We use BPEL templates to implement the protocols with BPEL processes at provider side.

A. The Coordination Model

The underlying coordination model is derived from WS-Coordination and also the XACML access control policy framework. We adapted this to the requirements of our coordination mechanism for localisation policy enforcement. The adapted coordination model uses two types of subcoordinators for process consumers and providers. In this scenario, a participant only interacts with its own coordinator type. The coordination model is defined as $< COOR, COOR_{context} >$ with $COOR = COOR^c \cup COOR^p$. Here, $coor^c \in COOR^c$ is a coordinator associated with the consumer and $coor^p \in COOR^p$ is a coordinator associated with the provider. $coor_{context} \in COOR_{context}$ captures the coordinaton context (involved services and locale definitions). Figure 5 illustrates how $coor^c$ and $coor^p$ interact in a coordination conversion. Protocol X and services X_c and X_p are instances in this coordination protocol.

1) The process consumer sends a create coordination context request to the activation service of $coor^c$. It will return an initialized localisation context $coor_{context}$ (Cc) that contains the identification, a service reference of the $coor^c$'s protocol service and other information for starting a conversation.
2) The process consumer then sends a process request to the provider or localisation process containing the $coor_{context}$.
3) The context $coor_{context}$ is extracted from the SOAP message and passed to protocol service X_p at $coor^p$.

Now, the protocol service X_c service reference is known to the protocol service X_p and theactual localisation-oriented communication between the participating services can be established.

4) The localisation coordination conversation ends with the completion of the process execution.

B. Process Activity Protocol

The process activity protocol defines a coordination type for coordination conversations based on the coordination model. A conversation of a localisation process is established for the coordination of the activities within the service consumer process. The model behind the coordination protocol is activity-centric, which means it can be applied to any localisation process irrespective of specific combination of localisation techniques applied. This coordination protocol applies to all activities of the processes to be managed on behalf of the client/consumer during execution. A coordination protocol consists of two main elements in ($ct \in \ coor_{context}$):

1) a message schema defines the message structure needed for services communication between consumer $COOR^c$ and provider $COOR^p$ for the extension element of the $COOR_{context}$.
2) a Finite State Machine (FSM) of $COOR^c$ and $COOR^p$ defining the actual localisation behaviour in an abstract model, described in more detail now.

The activity protocol defines runtime localisation management for localisation processes and the responsibilities of service providers and consumers in the acutal management and execution of localisation as a contract. This runtime mediation is formulated as an FSM defining the coordination protocol. There is an FSM for every activity in the processes that describes the behaviours of consumers and providers, $COOR^c$ and $COOR^p$, in the conversations. The idea behind this FSM design is to instrument the states into the process flow as these states determine which localisation policies are applicable.

The whole FSM is divided into two parts that are responsible for $COOR^c$ and $COOR^p$ separately. The $COOR^c$ FSM is a submachine of the FSM of $COOR^p$. Here, process providers only follow that part of the protocol that is actually defined for $COOR^p$. Similarly, consumers follow the FSM of $COOR^c$. As the FSM implementation is executed at the consumer and provider separately to achieve independence and, as already emphasised, the control of the consumer, $COOR^c$ needs sufficient information about process execution in order to execute process services at the provider side. The FSM of $COOR^c$ is defined for the submachine in the FSM of $COOR^p$, isolated from the process. Consequently, the protocol message schema only covers the activity information instead of the process state information. The execution of the $COOR^c$ FSM does only require information about the weaving request, which is instruments the original service with the localisation techniques to adapt the service. The execution of the $COOR^p$ FSM on the other hand does only require information about the weaving response. A consumer can customize the FSM of $COOR^c$ for itself without affecting the $COOR^p$ FSM and other process consumers, which is one of the novelties and selling points of our solution. Furthermore, this solution reduces complexity in the state machine execution for both sides. Each side does not need to know any implementation details of other participants for its own implementation.

The design with two separate FSMs reduces the number of states in the FSM of $COOR^p$, hence reducing the message exchange times required for coordination conversations. This reduces the performance overhead, which is generally caused by any communication between the services. Depending on current network properties between consumer and provider, the message exchange could reduce in delays and low performance, often not acceptable in runtime adaptation situations as the localisation here. Of course, we need to note here that this create an additional requirement for the consumer side, because of the $COOR^c$ FSM needs to be implemented at the client side. On the other hand, this allows different protocols for different localisation settings to be defined for $COOR^p$.

The FSM of $COOR^p$ specifies the protocol for $COOR^p$. The FSM of $COOR^c$ is specified in [7], [10]. The states reflect the status of the execution such as 'executing' or 'waiting' or 'completed'. A number of these states such as 'violated' or 'replacing' is necessary to deal with error situations that can occur in a distributed context where services or infrastructure can fail. These error states are based on common error handling strategies, as explained in [7], [10]. The FSM of $COOR^p$ is defined as a 5-tuple $(S, s_{start}, F, TA, \delta)$, where

- $S = S_g \cup S_{\neg g}$ is a set of states. S_l is a set of localisation states $\{s_{man\_val_{pre}}, s_{man\_val_{post}}, s_{handling_{pre}}, s_{handling_{pre}}, s_{cancelling}\}$ directly involved with process consumers or policies. The $S_{\neg l}$ is a set of non-localisation states $\{s_{start}, s_{violated_{pre}}, s_{executing}, s_{replacing}, s_{waiting}, s_{skipping}, s_{violated_{post}}, s_{compensating}, s_{com+rep}, s_{com+ign}, s_{completed}, s_{end}\}$ not directly involving consumers.
- $s_{start} \in S_{\neg l}$ is an initial state. The coordination can only be started by the process provider, i.e., not directly involved with the consumers.
- $F \subseteq S_{\neg l}$ is a set of final states $\{s_{end}\}$.
- $TA = TA_l \cup TA_{\neg l}$ is a set of input symbols for the localisation actions. TA_l is a set of localisation transaction actions $\{ta_{violate}, ta_{validated}, ta_{ignore}, ta_{replace}, ta_{skip}, ta_{cancel}, ta_{compensate}, ta_{retry}, ta_{com+ign}, ta_{com+rep}\}$ expected from process consumers, again to handle errors. $TA_{\neg l}$ is a set of transaction actions, which are not expected from process consumers $\{0,\ 1\}$. The input stream of the FSM regarding $TA_{\neg l}$ is decided by the process providers based on the process state information that is not covered by the FSM (note that FSM is only activity-scoped).
- δ is a transition system $\delta : S \times TA \rightarrow S$, see transition graph in Figure 7.

C. Coordination Implementation and BPEL Instrumentation

The coordination protocol needs to be implemented to enable coordination. The difficulty is on the provider side, since all activities within a localisation process need to comply with the defined protocol during the BPEL execution.

We designed a set of templates for BPEL to avoid platform dependency, i.e., to allow this to be applied to different BPEL

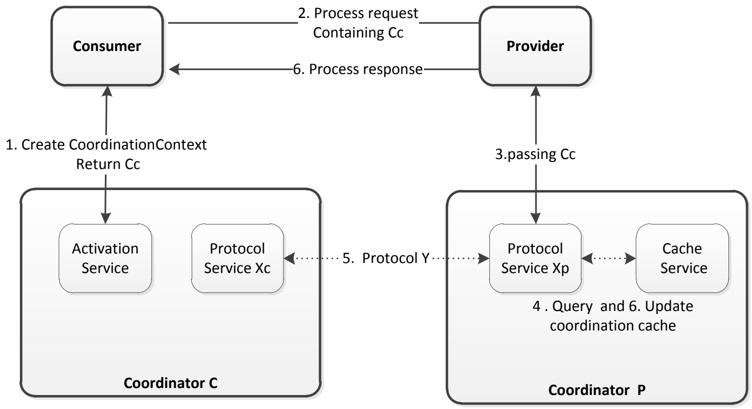


Figure 5: Policy Coordination Architecture.

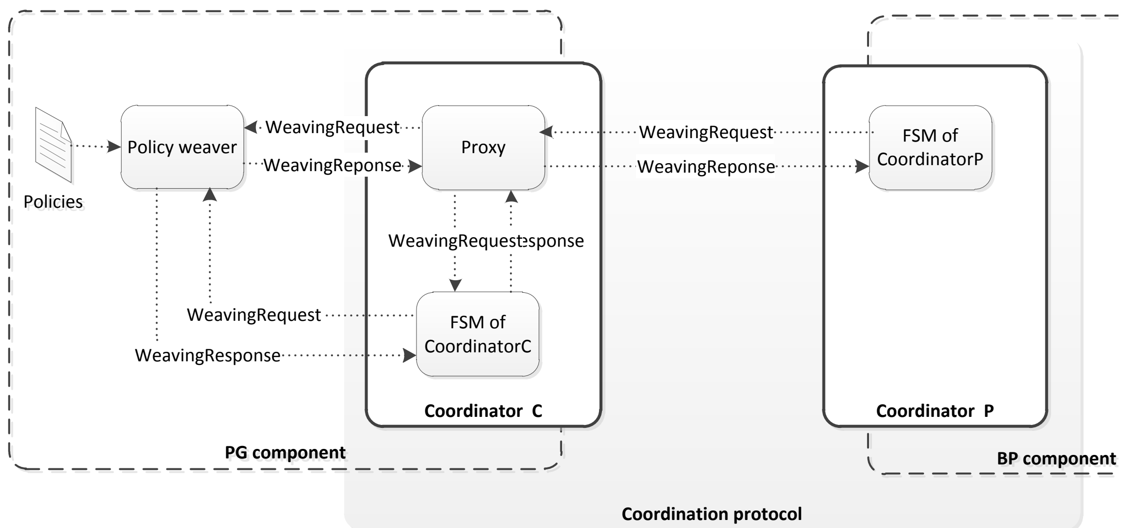


Figure 6: Message flow diagram

engines. The protocol needs to be implemented with a BPEL process as a $coor^p$ for activities. The process contains the flow logic to be executed and can be driven by protocol messages. A process instance, i.e., not the BPEL process, is associated with a coordination conversation belonging to a consumer to enable user-centric customisation.

In order to separate concers, we divide the FSM of $COOR_p$ into two sections. The first is process-independent, i.e., does not require awareness of the process states. This part of the FSM implementation is wrapped up in the main BPEL process. The second part continues the FSM to the end state of the main process. The first part can be implemented as BPEL processes, but as processes separate from the main process. Using such a hybrid approach, we can achieve a platform-independent approach that also keeps the main BPEL code simple. As a limitation we need to note that the BPEL processes here are protocol-specific. We can use the BPEL transaction scope concept in order to implement the FSMs with BPEL as long-running transactions (LRTs). These LRTs in BPEL focus on scopes and these scopes can even be nested. That means that when a fault occurs, all previously committed activities can either be compensated within the faulty process, or compensated as an activity in the parent process.

A template approach allows for easier management. Two templates for BPEL process development reduce the protocol implementation effort. A template defines the abstract skeleton of an algorithm. One or more of the algorithm steps can be overridden by subclasses allowing to define differing localisation behaviours by the consumer while at the same time ensuring that the overall protocol is followed. We extract the first FSM section as the non-transactional requirement FSM for localisation process activities. The second section is then an extension for activities to support transactions. The FSM is divided into two implementation parts with two respective templates: wrapper service template and main process template.

This process template is an implementation of the second part of the FSM containing activity states from $s_{completed}$ to the s_{end} state. When the process is in a cancelling state, the previous successfully executed activities can be compensated if that is necessary. The template is designed with an activity scope and a process scope, respectively.

Figure 8 shows the BPEL template for the activity scope associated with activity states is also needed. The template for each activity is a separate scope. The two services inside the template are highlighted by grey boxes. The first service is the wrapper service for the first part of the FSM implementation.

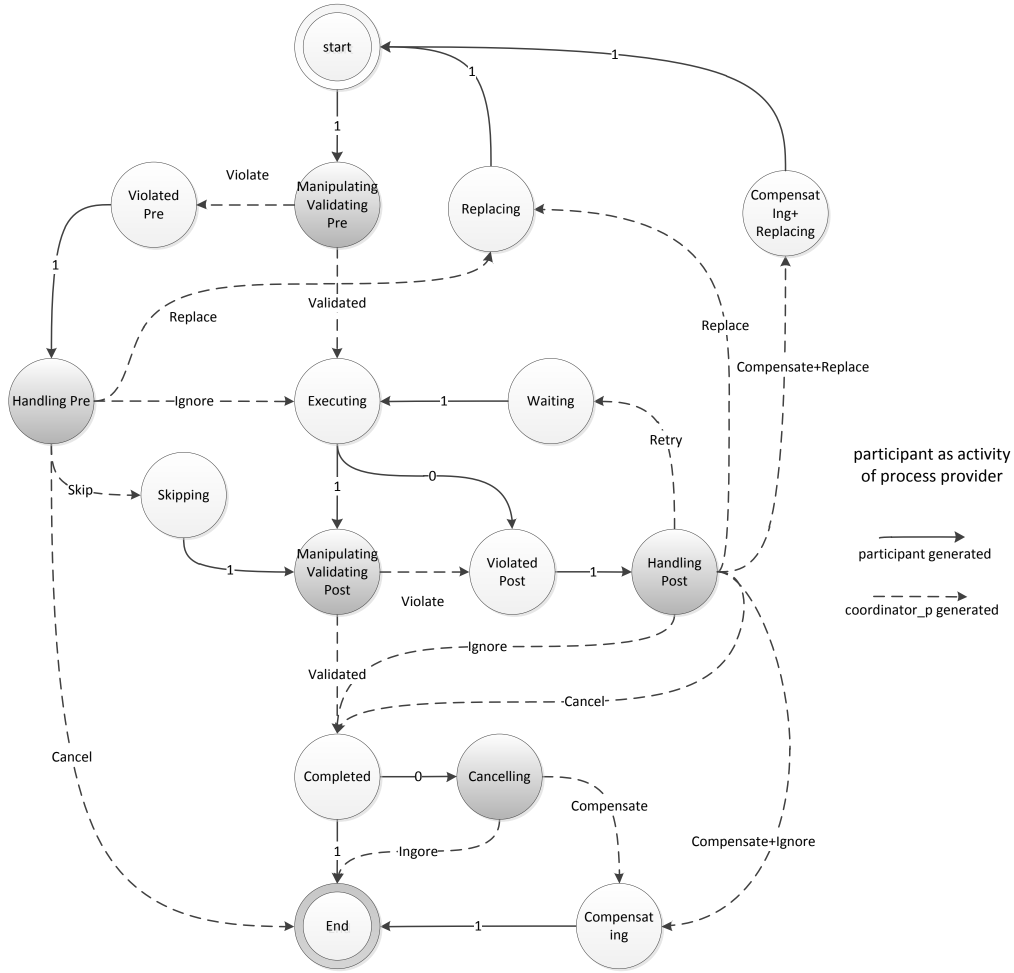


Figure 7: Transition graph for FSM for $Coor^p$

The required variables are passed into the BPEL process by a BPEL <*assign*> activity. With the BPEL <*if*> control structure, a <*throw*> activity throws a defined fault if the *comp* variable is set to false. An attached BPEL <*catchAll*> handler catches the fault and marks this scope as faulty. The BPEL <*compensationHandler*> attachment is only triggered by a successful scope if the process in a cancelling state. In this situation, e.g., if the execution state $s_{executing}$ is skipped in the first FSM part, the compensation handler for the activity scope is then triggered. The scope is marked as faulty instead. The last <*if*> conditional control structure marks the process as being in cancellation status. In these cases throws a defined fault and to be caught in a <*catchAll*> handler defined in the process scope template. Thus, the <*compensationHandler*> handler at the corrwesponding activity scope is triggered. The process activities are executed from state $s_{completed}$ to state $s_{cancelling}$ if necessary. A utility service within the <*compensationHandler*> moves on from the activity state $s_{cancelling}$ to the state s_{end}.

Finally, Figure 9 shows the BPEL template for the process scope. All process activities are within a process scope – this is associated to a <*catchAll*> handler. If a defined fault for process cancellation is caught by the handler with the process scope, all <*compensationHandler*>*s* of activity templates of fault-free activities are executed in reverse order. Activities in the $s_{completed}$ state will transfer to the cancellation state $s_{cancelling}$. In case of nested processes, the parent would handle the situation. The violation handling would depends on the fault policy defined in the parent process. We have not covered these fault aspect here in details, as our focus was on the core localisation activities. The provision of fault handling is necessary to provide a credible solution architecture.

VI. Implementation and Evaluation

The localisation platform presented here was fully implemented in a Java prototype for the localisation techniques, combined with the coordination solution based on WS-Coordination and BPEL, that aims at studying the feasibility of the conceptual solution.

We carried out an experimental evaluation. The objectives are explanatory aiming to demonstrate and confirm acceptable usability based on technical criteria. It shall be assessed based on the following criteria here: Performance and Extensibility. These criteria have different effects on the end-user experience of the product. These criteria are key performance indicators

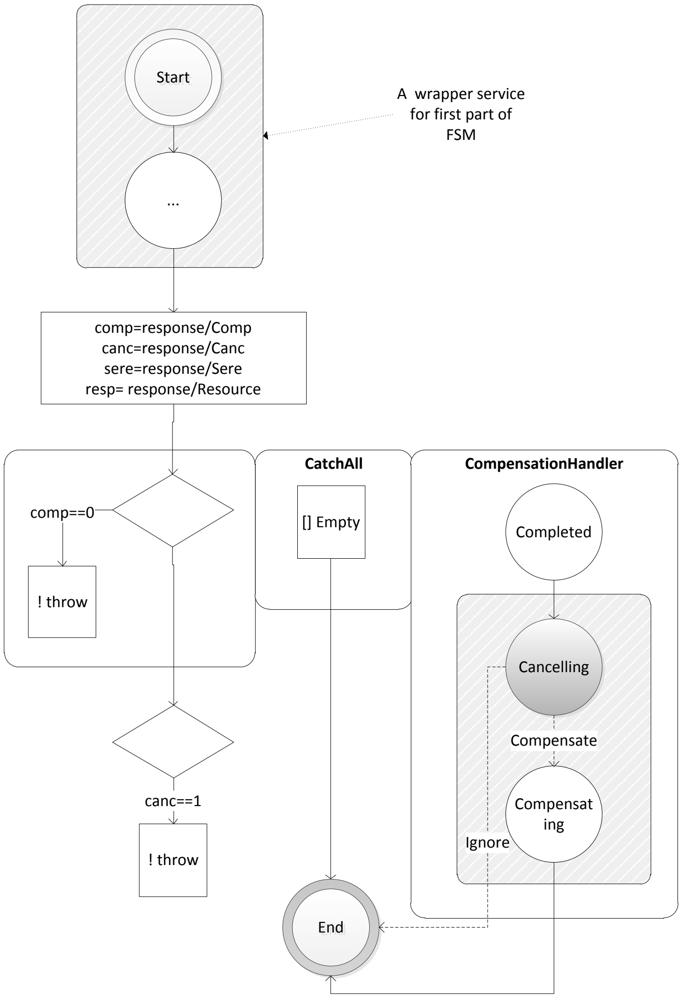


Figure 8: Activity scope BPEL template

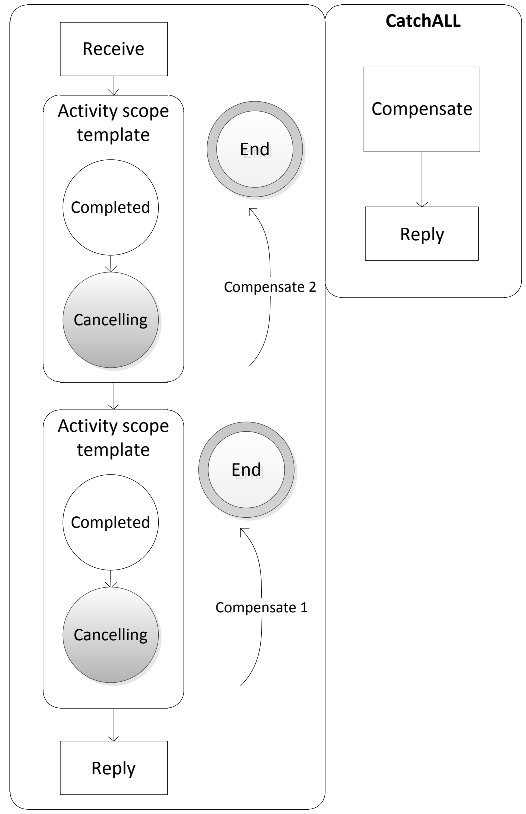


Figure 9: Process scope BPEL template

(KPI) and critical success factors (CSF) of the localisation platform described. We analyse the resulting data quantitatively.

Please note that other relevant aspects of the solutions, for instance the performance of the coordination solution in a generic form have been presented elsewhere [7], [10]. These have established an overhead of around ten per cent for the coordination framework - however, the ten per cent essentially come into effect if fault handling is needed, as the templates in the previous section indicate. As said, the focus here is on the localisation services themselves that are facilitated through the coordination platform.

Another aspect is accuracy. This could vary across the different localisation dimensions. While currency or unit conversions, even if dynamically done and not pre-computed are simple and can expect to be fully accurate, standard-based mappings depend on the quality of expert input and language translation on the machine translation quality. As in particular the language translation aspects have not been deeply addressed here, a comprehensive analysis is not possible.

Performance. Poor performance often tends to affect software exponentially as multiples of users consume a service at the same time. The core question here is the overhead created by adding localisation dynamically to service provisioning. Our results show an acceptable overhead of 10-15 % additional execution time for fully localised services (i.e., localisation involving different localisation aspects). The overhead is still low compared to network latency and the average service execution time [7]. As the application deals with multiple users, the latency would increase due to extra loads placed on the platforms services. This makes latency one of the key concerns of the project. Latency is also an area to be assessed as adding the localisation platform to the workflow of an existing process has the potential to add to processing delays. This delay exists due to time required to compute and also the time to initialise the various variables. The propagation latency is displayed in Table IV below. The figures are based on randomly distributed service calls to external and internal localisation services (e.g., external currency conversion or internal unit conversion or hybrid conversions as for translations) based on stock market services (NASDAQ, FTSE). For a number of localisation policies, the individual response times have been aggregation and normalised. Note that figures can be affected by environmental changes or the locale we are transforming from and the locale we are transforming to.

TABLE IV: Latency Table - Localisation of Service

Service	Prior (μs)	Post (μs)	Δt (μs)
NASDAQ	132	182	50
FTSE	110	152	42

As a general strategy, we have aimed to improve performance of the prototype by using pre-translated aspects through stored mappings, e.g., for currency conversions and standard translations, which suggests that further optimisations are possible.

Scalability. A related concern is scalability of software becomes more important when a service may have large multiples of users, which can be the case if several clients use the localisation framework at the same time. The performance evaluation has been carried out for a single user to determine the overhead of localisation for a single service call. Scalability has not been empirically addressed for this phase of research and will be evaluated in later prototypes that will implement a more scalable base architecture.

- Some components of the platform would require modification to effectively allow the infrastructure to vertically scale-up or scale-out efficiently. Solutions here are stateless programming and data externalisation. Through our rule base, and the suggested pre-translation repositories some suitable architectural decision in this direction have already been made.
- Horizontal scalability - i.e., the addition of more localisation concerns - is conceptually easily supported by the modular mediator architecture, which we will address further below in the extensibility context from an implementation view.

An interesting model to investigate the scalability is a tuple space-based coordination approach [11]–[13], which would allow a flexible and elastic assignment of localisation services to multiple requests. Work by Creaner and Pahl [11] suggest a good scalability poential through tuple-space for coordination.

Extensibility. Extensibility becomes important when dealing with complete platforms like a localisation platform. During an initial development, it is often the case that features need to be included due to various constraints. In the case of the localisation platform described here, some localisation services where not developed, some of which include a service to handle taxation. However, the platform was designed to be extendable. At a platform level, this allows for the addition of further services and the support for more locales. While we do not have empirical evidence, the extensibility comes as a common property of ontologies and semantic Web rule languages as easily extensible frameworks.

VII. Related Work

We provide a different view and perspective on the subject compared to other publications on service adaptation [6], [14], [15], [17] that look at adaptation from a technology perspective. The area of localisation in its wider adaptivity and customisation sense has been worked on in various EU-supported research projects, such as SOA4ALL [18] and 4Caast [19]. These projects address end-user adaptation through the use of generic semantic models [20], [21]. Areas such as software coordination are also covered. The mOSAIC project adds multi-cloud provision to the discussion. Our framework however is modular and extensible and aims to provide a one-stop shop for all localisation methods. The 4CaaSt contributors have work on a semantic platform for service adaptions. The semantic mobility channel proposed by Cantera et al. links semantic technologies with recent service platforms. While the platform technology proposed there is applicable and provides for instance label-based RDF adaptation, the specific dimensions we identified go beyond their proposal.

The platform that is described here addresses the need for dynamic localisation of various artefacts by use of a translation memory and a set of logical rules. Software Localisation refers to human consumption of data produced by the software - namely messages and dialogues. Our focus is on the localisation of the service level. Service internationalisation is supported by the W3C Service Internationalisation activity [22], [23]. Adaptation and Integration of services based on locales and using a translation memory with rules and mappings is new [17]. The problem of multi-tenancy is a widespread issue in the area of cloud computing [7]. This is an area where a lot of research is being invested in order to provide a platform for different users with different business needs to be kept separate and their data to be kept private. Semantics involves the matching of services with various locales using mappings and rule-based system [15], [16], [24], [25].

There are implementations that can perform localisation operations on web services [26]. The use of some of these, however, is restricted due to their nature. Some of the other implementations require a specific Integrated Development Environment or specific proprietary libraries. They also typically enable localisation at compile time - the proposed implementation in this paper is to enable service localisation at run time. IBM has presented a static localisation solution suitable for web services using its WebSphere platform [26], which requires the WSDL files to be generated within the Integrated Development Environment prior to deployment. This differs from our proposed localisation platform as our solution aims to perform transformations between locales dynamically.

VIII. Conclusion and Future Work

Service localisation falls into the service personalisation and adaptation context. There are particular engineering methods and tools that can be employed to allow services to be adapted to different locales. A Service Localisation implementation should allow for automatically adjusting and adapting services to the requesters' own locales defined by language or regulatory environment. We have presented a modular implementation. Localisation hence provides a mechanism to widen a service provider's target market by enabling multi-locale solutions. The easiest solution is for a service provider to provide a 'mediator' service that could act as middleware between a requester and the service provider. In order to enhance the dynamic evolution of localisation settings, our coordination infrastructure allows a client-side driven management of localisation settings.

By allowing services to be localised, we are enabling the provision of multi-locale services to create interoperable service ecosystems (such as clouds). Due to the nature of third-party services, it is more intuitive for service localisation to be performed dynamically through the use of a mediator service, controlled by the client and enacted by the provider. Service localisation thus enables higher availability of services through its use of innovative interfacing. This type of localisation would be value-add for a company, which may not have the resources to perform localisation in-house. It also allows service consumers more influence on the type of localisation and frequency of localisation changes.

The objectives of Service Localisation have been presented in terms of three aspects. Firstly, presented was a conceptual framework, which demonstrated key motivational reasons for developing a multi-locale support framework. The second part presented a modular platform, which is extensible to allow the support of further localisable artefacts. The platform that was implemented was using Java libraries was discussed as this programming solution copes well with the problem of extensibility. The third part introduces the coordination platform to coordinate conversations between different service providers and consumers and manage potential failure.

The proposed service localisation fills a gap. Software adaptation has looked into adapting for instances services in terms of their user's interface needs such as data types and formats. The two focal localisation concerns lingual and regulatory add new perspectives to this area of research. A different activity is the Web services internationalisation effort, which looks into basic localisation concerns such as units, currency or the format of dates. Our localisation solution includes these (as we have demonstrated with the currency aspect), but expands these into a comprehensive framework.

The context of adaptation and translations/mappings used to facilitate this is a broad field. Our aim here was to integrate difference concerns into a coherent localisation framework. This relies on individual mappings. As part of our future work, we aim to add a semantic layer, which would support wider localisation concerns in an integrated format. Firstly, it would allow more reliable translations for non-trivial concerns if overarching ontologies were present. Secondly, the different concerns themselves could be integrated by determining inter-dependencies. Another direction of future research would be to look into composition and specifically the behaviour of individual service localisation in for instance service orchestrations or other coordination models (e.g., tuple spaces as suggested above to deal more specifically with scalability problems) [13], [27]. Translation is an aspect that deserves more attention. Translation of technical content based on reduced-context machine translation techniques is an avenue [29]. Furthermore, techniques are needed to facilitate reliable translations between technical content representations, e.g., formalised, technical content like ontologies, service API and service models [28].

References

[1] L. Collins and C. Pahl, "A service localisation platform," The Fifth International Conferences on Advanced Service Computing (Service Computation 2013), IARIA, pp. 6-12, 2013.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," Communications of the ACM, vol. 53(4), pp. 50-58, 2010.

[3] W. Voorsluys, J. Broberg, and R. Buyya, "Cloud computing: principles and paradigms." Hoboken (NJ), USA: John Wiley and Sons, 2011.

[4] K. Chen and W. Zheng, "Cloud computing: system instances and current research," International Conference on Future Networks, ICFN '10, pp. 88-92, 2010.

[5] P. Fingar, "Cloud computing and the promise of on-demand business innovation," InformationWeek, July 13, 2009.

[6] C. Pahl, "Cloud service localisation," European Conference on Service-Oriented and Cloud Computing (ESOCC 2012), pp. 138-153, 2012.

[7] M.X. Wang, K.Y. Bandara, and C. Pahl, "Process as a service distributed multi-tenant policy-based process runtime governance," International Conference on Services Computing (SCC'10), IEEE, pp. 578-585, 2010.

[8] H. Weigand, W. van den Heuvel, and M. Hiel, "Rule-based service composition and service-oriented business rule management," Proceedings of the International Workshop on Regulations Modelling and Deployment (ReMoD'08), pp. 1-12, 2008.

[9] Y. Wu and P. Doshi, "Making BPEL flexible and adapting in the context of coordination constraints using WS-BPEL," *Intl Conf on Services Computing*, pp. 423 - 430, 2008.

[10] M.X. Wang, K.Y. Bandara, and C. Pahl, "Integrated constraint violation handling for dynamic service composition," IEEE Intl Conf on Services Computing, pp. 168-175, 2009.

[11] G. Creaner and C. Pahl, "Flexible coordination techniques for dynamic cloud service collaboration," Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions, J. Cubo and G. Ortiz, Eds. Hershey (PA), USA: IGI Global, pp. 239-252, 2012.

[12] E.-E. Doberkat,W. Hasselbring, W. Franke, U. Lammers, U. Gutenbeil, and C. Pahl, "ProSet - a language for prototyping with sets," International Workshop on Rapid System Prototyping, IEEE Press, pp. 235-248, 1992.

[13] C. Pahl, "Dynamic adaptive service architecturetowards coordinated service composition," European Conference on Software Architecture (ECSA'2010), pp. 472-475, 2010.

[14] L. Baresi and S. Guinea, "Self-supervising BPEL processes," IEEE Transactions on Software Engineering, vol. 37(2), pp. 247-263, 2011.

[15] K. Fujii and T. Suda, "Semantics-based context-aware dynamic service composition," ACM Transactions on Autonomous and Adaptive Systems, vol. 4(2), pp. 12-21, 2009.

[16] R. Barrett, L. M. Patcas, C. Pahl, and J. Murphy, "Model driven distribution pattern design for dynamic Web service compositions," International Conference on Web Engineering (ICWE 2006), ACM Press, pp. 129-136, 2006.

[17] H. Truong and S. Dustdar, "A survey on context-aware web service systems," Intl Journal of Web Information Systems, vol. 5(1), 2009, pp. 5-31.

[18] SOA4All Consortium, "Service oriented architectures for all," EU FP7 IST Project, 2012.

[19] 4CaaSt Consortium, "Building the PaaS cloud of the future," EU FP7 IST Project, 2013.

[20] C. Pahl, S. Giesecke, and W. Hasselbring, "An ontology-based approach for modelling architectural styles," European Conference on Software Architecture (ECSA'2007), pp. 60-75, 2007.

[21] C. Pahl, S. Giesecke, and W. Hasselbring, "Ontology-based modelling of architectural styles," Information and Software Technology, vol. 1(12), pp. 1739-1749, 2009.

[22] A. Phillips, "Web services and internationalization," W3C Whitepaper, 2005.

[23] W3C, "Web services internationalization usage scenarios," W3C Internationalisation, Working Group Note, 2004.

[24] D. Anastasiou, "The impact of localisation on semantic web standards," European Journal of ePractice, vol. 12, pp. 42-52, 2011.

[25] K.Y. Bandara, M.X. Wang, and C. Pahl, "Dynamic integration of context model constraints in web service processes," International Software Engineering Conference (SE'2009), IASTED, 2009.

[26] IBM, "Developing internationalized Web services with WebSphere Business Integration Server Foundation V5.1," IBM Developer Technical Journal, 2010.

[27] C. Pahl, "Layered ontological modelling for Web service-oriented model-driven architecture," European Conference on Model-Driven Architecture - Foundations and Applications (ECMDA'05), Springer, pp. 88-102, 2005.

[28] J.v. Genabith, "Metaphors, Logic and Type Theory," Metaphor and Symbol, vol. 16(1/2), pp. 42-57, 2001.

[29] J.v. Genabith and R. Crouch, "Dynamic and Underspecified Semantics for LFG," Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach, Cambridge: USA: MIT Press, pp. 209-260, 1999.

A Scalable Backward Chaining-based Reasoner for a Semantic Web

Hui Shi
Department of Management and Information Sciences
University of Southern Indiana
Evansville, USA
hshi@cs.odu.edu

Kurt Maly, Steven Zeil
Department of Computer Science
Old Dominion University
Norfolk, USA
{maly, zeil}@cs.odu.edu

***Abstract* — In this paper we consider knowledge bases that organize information using ontologies. Specifically, we investigate reasoning over a semantic web where the underlying knowledge base covers linked data about science research that are being harvested from the Web and are supplemented and edited by community members. In the semantic web over which we want to reason, frequent changes occur in the underlying knowledge base, and less frequent changes occur in the underlying ontology or the rule set that governs the reasoning. Interposing a backward chaining reasoner between a knowledge base and a query manager yields an architecture that can support reasoning in the face of frequent changes. However, such an interposition of the reasoning introduces uncertainty regarding the size and effort measurements typically exploited during query optimization. We present an algorithm for dynamic query optimization in such an architecture. We also introduce new optimization techniques to the backward-chaining algorithm. We show that these techniques together with the query-optimization reported on earlier, will allow us to actually outperform forward-chaining reasoners in scenarios where the knowledge base is subject to frequent change. Finally, we analyze the impact of these techniques on a large knowledge base that requires external storage.**

Keywords-semantic web; ontology; reasoning; query optimization; backward chaining.

I. INTRODUCTION

Consider a potential chemistry Ph.D. student who is trying to find out what the emerging areas are that have good academic job prospects. What are the schools and who are the professors doing groundbreaking research in this area? What are the good funded research projects in this area? Consider a faculty member who might ask, "Is my record good enough to be tenured at my school? At another school?" It is possible for these people each to mine this information from the Web. However, it may take a considerable effort and time, and even then the information may not be complete, may be partially incorrect, and would reflect an individual perspective for qualitative judgments. Thus, the efforts of the individuals neither take advantage of nor contribute to others' efforts to reuse the data, the queries, and the methods used to find the data. We believe that some of these qualitative descriptors such as "groundbreaking research in data mining" may come to be accepted as meaningful if they represent a consensus of an appropriate subset of the community at large. However, even in the absence of such sharing, we believe the expressiveness of user-defined qualitative descriptors is highly desirable.

The system implied by these queries is an example of a semantic web service where the underlying knowledge base covers linked data about science research that are being harvested from the Web and are supplemented and edited by community members. The query examples given above also imply that the system not only supports querying of facts but also rules and reasoning as a mechanism for answering queries.

A key issue in such a semantic web service is the efficiency of reasoning in the face of large scale and frequent change. Here, scaling refers to the need to accommodate the substantial corpus of information about researchers, their projects and their publications, and change refers to the dynamic nature of the knowledge base, which would be updated continuously [1].

In semantic webs, knowledge is formally represented by an ontology as a set of concepts within a domain, and the relationships between pairs of concepts. The ontology is used to model a domain, to instantiate entities, and to support reasoning about entities. Common methods for implementing reasoning over ontologies are based on First Order Logic, which allows one to define rules over the ontology. There are two basic inference methods commonly used in first order logic: forward chaining and backward chaining [2].

A question/answer system over a semantic web may experience changes frequently. These changes may be to the ontology, to the rule set or to the instances harvested from the web or other data sources. For the examples discussed in our opening paragraph, such changes could occur hundreds of times a day. Forward chaining is an example of data-driven reasoning, which starts with the known data in the knowledge base and applies modus ponens in the forward direction, deriving and adding new consequences until no more inferences can be made. Backward chaining is an example of goal-driven reasoning, which starts with goals from the consequents, matching the goals to the antecedents to find the data that satisfies the consequents. As a general rule forward chaining is a good method for a static knowledge base and backward chaining is good for the more dynamic cases.

The authors have been exploring the use of backward chaining as a reasoning mechanism supportive of frequent changes in large knowledge bases. Queries may be composed of mixtures of clauses answerable directly by access to the knowledge base or indirectly via reasoning applied to that base. The interposition of the reasoning introduces uncertainty regarding the size and effort associated with resolving individual clauses in a query. Such uncertainty poses a challenge in query optimization, which typically relies upon the accuracy of these estimates. In this paper, we describe an approach to dynamic optimization that is effective in the presence of such uncertainty [1].

In this paper, we will also address the issue of being able to scale the knowledge base beyond the level standard backward-chaining reasoners can handle. We shall introduce new optimization techniques to a backward-chaining algorithm and shall show that these techniques, together with query-optimization, will allow us to actually outperform forward-chaining reasoners in scenarios where the knowledge base is subject to frequent change.

Finally, we explore the challenges posed by scaling the knowledge base to a point where external storage is required. This raises issues about the middleware that handles external storage, how to optimize the amount of data and what data are to be moved to internal storage.

In Section II, we provide background material on the semantic web, reasoning, and database querying. Section III gives the overall query-optimization algorithm for answering a query. In Section IV, we report on experiments comparing our new algorithm with a commonly used backward chaining algorithm. Section V introduces the optimized backward-chaining algorithm and Section VI provides details on the new techniques we have introduced to optimize performance. A preliminary evaluation of these techniques on a smaller scale, using in-memory storage, is reported in a separate paper [3]. In Section VII, we describe the issues raised when scaling to an externally stored knowledge base, evaluate the performance of our query optimization and reasoner optimizations in that context, and perform an overall comparison with different data base implementations.

II. RELATED WORK

A number of projects (e.g., Libra [4][5], Cimple [6], and Arnetminer [7]) have built systems to capture limited aspects of community knowledge and to respond to semantic queries. However, all of them lack the level of community collaboration support that is required to build a knowledge base system that can evolve over time, both in terms of the knowledge it represents as well as the semantics involved in responding to qualitative questions involving reasoning.

Many knowledge bases [8-11] organize information using ontologies. Ontologies can model real world situations, can incorporate semantics, which can be used to detect conflicts and resolve inconsistencies, and can be used together with a reasoning engine to infer new relations or proof statements.

Two common methods of reasoning over the knowledge base using first order logic are forward chaining and backward chaining [2]. Forward chaining is an example of data-driven reasoning, which starts with the known data and applies modus ponens in the forward direction, deriving and adding new consequences until no more inferences can be made. Backward chaining is an example of goal-driven reasoning, which starts with goals from the consequents matching the goals to the antecedents to find the data that satisfies the consequents. Materialization and query-rewriting are inference strategies adopted by almost all of the state of the art ontology reasoning systems. Materialization means pre-computation and storage of inferred truths in a knowledge base, which is always executed during loading the data and combined with forward-chaining techniques. Query-rewriting means expanding the queries, which is always executed during answering the queries and combine with backward-chaining techniques.

Materialization and forward chaining are suitable for frequent computation of answers with data that are relatively static. OWLIM [12] and Oracle 11g [13], for example implement materialization. Query-rewriting and backward chaining are suitable for efficient computation of answers with data that are dynamic and infrequent queries. Virtuoso [14], for example, implements a mixture of forward-chaining and backward-chaining. Jena [15] supports three ways of inferencing: forward-chaining, limited backward-chaining and a hybrid of these two methods.

In conventional database management systems, query optimization [16] is a function to examine multiple query plans and selecting one that optimizes the time to answer a query. Query optimization can be static or dynamic. In the Semantic Web, query optimization techniques for the common query language, SPARQL [17][18], rely on a variety of techniques for estimating the cost of query components, including selectivity estimations [19], graph optimization [20], and cost models [21]. These techniques assume a fully materialized knowledge base.

Benchmarks evaluate and compare the performances of different reasoning systems. The Lehigh University Benchmark (LUBM) [22] is a widely used benchmark for evaluation of Semantic Web repositories with different reasoning capabilities and storage mechanisms. LUBM includes an ontology for university domain, scalable synthetic OWL data, and fourteen queries.

III. DYNAMIC QUERY OPTIMIZATION WITH AN INTERPOSED REASONER

A query is typically posed as the conjunction of a number of clauses. The order of application of these clauses is irrelevant to the logic of the query but can be critical to performance.

In a traditional data base, each clause may denote a distinct probe of the data base contents. Easily accessible information about the anticipated size and other characteristics of such probes can be used to facilitate query optimization.

The interposition of a reasoner between the query handler and the underlying knowledge base means that not all clauses will be resolved by direct access to the knowledge base. Some will be handed off to the reasoner, and the size and other characteristics of the responses to such clauses cannot be easily predicted in advance, partly because of the expense of applying the reasoner and partly because that expense depends upon the bindings derived from clauses already applied. If the reasoner is associated with an ontology, however, it may be possible to relieve this problem by exploiting knowledge about the data types introduced in the ontology.

In this section, we describe an algorithm for resolving such queries using dynamic optimization based, in part, upon summary information associated with the ontology. In this algorithm, we exploit two key ideas: 1) a greedy ordering of the proofs of the individual clauses according to estimated sizes anticipated for the proof results, and 2) deferring joins of results from individual clauses where such joins are likely to result in excessive combinatorial growth of the intermediate solution.

We begin with the definitions of the fundamental data types that we will be manipulating. Then we discuss the algorithm for answering a query. A running example is provided to make the process more understandable.

We model the knowledge base as a collection of triples. A triple is a 3-tuple (x,p,y) where x, p, and y are URIs or constants and where p is generally interpreted as the identifier of a property or predicate relating x and y. For example, a knowledge base might contains triples

(Jones, majorsIn, CS), (Smith, majorsIn, CS),
(Doe, majorsIn, Math), (Jones, registeredIn, Calculus1),
(Doe, registeredIn, Calculus1).

A QueryPattern is a triple in which any of the three components can be occupied by references to one of a pool of entities considered to be variables. In our examples, we will denote variables with a leading '?'. For example, a query pattern denoting the idea "Which students are registered in Calculus1?" could be shown as

(?Student,registeredIn,Calculus1).

A query is a request for information about the contents of the knowledge base. The input to a query is modeled as a sequence of QueryPatterns. For example, a query "What are the majors of students registered in Calculus1?" could be represented as the sequence of two query patterns

[(?Student,registeredIn,Calculus1),
(?Student, majorsIn, ?Major)].

The output from a query will be a QueryResponse. A QueryResponse is a set of functions mapping variables to values in which all elements (functions) in the set share a common domain (i.e., map the same variables onto values). Mappings from the same variables to values can be also referred to as variable bindings. For example, the QueryResponse of query pattern (?Student, majorsIn, ?Major) could be the set

{{?Student => Jones, ?Major=>CS},
{?Student => Smith, ?Major=>CS },
{?Student => Doe, ?Major=> Math }}.

The SolutionSpace is an intermediate state of the solution during query processing, consisting of a sequence of (preliminary) QueryResponses, each describing a unique domain. For example, the SolutionSpace of the query "What are the majors of students registered in Calculus1?" that could be represented as the sequence of two query patterns as described above could first contain two QueryResponses:

[{{?Student => Jones, ?Major=>CS},
{?Student => Smith, ?Major=>CS },
{?Student => Doe, ?Major=> Math }},
{{?Student => Jones},{?Student => Doe }}]

Each Query Response is considered to express a constraint upon the universe of possible solutions, with the actual solution being intersection of the constrained spaces. An equivalent Solution Space is therefore:

[{{?Student => Jones, ?Major=>CS},
{?Major => Math, ?Student =>Doe}}],

Part of the goal of our algorithm is to eventually reduce the Solution Space to a single Query Response like this last one.

Fig. 1 describes the top-level algorithm for answering a query. A query is answered by a process of progressively restricting the SolutionSpace by adding variable bindings (in the form of Query Responses). The initial space with no bindings ❶ represents a completely unconstrained SolutionSpace. The input query consists of a sequence of query patterns.

We repeatedly estimate the response size for the remaining query patterns ❷, and choose the most restrictive pattern ❸ to be considered next. We solve the chosen pattern by backward chaining ❹, and then merge the variable bindings obtained from backward chaining into the SolutionSpace ❺

```
QueryResponseanswerAQuery(query: Query)
{
   // Set up initial SolutionSpace
   SolutionSpacesolutionSpace = empty; ❶
   // Repeatedly reduce SolutionSpace by
   //applying the most restrictive pattern
   while (unexplored patterns remain
     in the query) {
     computeEstimatesOfReponseSize
        (unexplored patterns); ❷
     QueryPattern p = unexplored pattern
        With smallest estimate; ❸
     // Restrict SolutionSpace via
     // exploration of p
     QueryResponseanswerToP =
        BackwardChain(p); ❹
     solutionSpace.restrictTo (
        answerToP); ❺
     }
    return solutionSpace.finalJoin();❻
}
```

Figure 1. Answering a Query.

via the restrictTo function, which performs a (possibly deferred) join as described later in this section.

When all query patterns have been processed, if the SolutionSpace has not been reduced to a single Query Response, we perform a final join of these variable bindings into single one variable binding that contains all the variables involved in all the query patterns ❻. The finalJoin function is described in more detail later in this section.

The estimation of response sizes in ❷ can be carried out by a combination of 1) exploiting the fact that each pattern represents that application of a predicate with known domain and range types. If these positions in the triple are occupied by variables, we can check to see if the variable is already bound in our SolutionSpace and to how many values it is bound. If it is unbound, we can estimate the size of the domain (or range) type, 2) accumulating statistics on typical response sizes for previously encountered patterns involving that predicate. The effective mixture of these sources of information is a subject for future work.

For example, suppose there are 10,000 students, 500 courses, 50 faculty members and 10 departments in the knowledge base. For the query pattern (?S takesCourse ?C), the domain of takesCourse is Student, while the range of takesCourse is Course. An estimate of the numbers of triples matching the pattern (?S takesCourse ?C) might be 100,000 if the average number of courses a student has taken is ten, although the number of possibilities is 500,000.

By using a greedy ordering ❸ of the patterns within a query, we hope to reduce the average size of the SolutionSpaces. For example, suppose that we were interested in listing all cases where any student took multiple courses from a specific faculty member. We can represent this query as the sequence of the patterns in Table I. These clauses are shown with their estimated result sizes indicated in the subscripts. The sizes used in this example are based on one of our LUBM [22] prototypes.

To illustrate the effect of the greedy ordering, let us assume first that the patterns are processed in the order given. A trace of the answerAQuery algorithm, showing one row for each iteration of the main loop is shown in Table II. The worst case in terms of storage size and in terms of the size of the sets being joined is at the join of clause 2, when the join of two sets of size 100,000 yields 1,000,000 tuples.

Now, consider the effect of applying the same patterns in ascending order of estimated size, shown in Table III. The worst case in terms of storage size and in terms of the size of the sets being joined is at the final addition of clause 2, when a set of size 100,000 is joined with a set of 270. Compared to Table II, the reduction in space requirements and in time required to perform the join would be about an order of magnitude.

TABLE I. EXAMPLE Query 1

Clause #	QueryPattern	Query Response
1	?S1 takesCourse ?C1	$\{(?S1=>s_i,?C1=>c_i)\}_{i=1..100,000}$
2	?S1 takesCourse ?C2	$\{(?S1=>s_j,\ ?C2=>c_j)\}_{j=1..100,000}$
3	?C1 taughtBy fac1	$\{(?C1=>c_j)\}_{j=1..3}$
4	?C2taughtBy fac1	$\{(?C2=>c_j)\}_{j=1..3}$

TABLE II. TRACE OF JOIN OF CLAUSES IN THE ORDER GIVEN

Clause Being Joined	Resulting SolutionSpace
(initial)	[ ]
1	$[\{(?S1=>s_i,\ ?C1=>c_i)\}_{i=1..100,000}]$
2	$[\{(?S1=>s_i,\ ?C1=>c_i,\ ?C2=>c_i)\}_{i=1..1,000,000}]$ (based on an average of 10 courses / student)
3	$[\{(?S1=>s_i,\ ?C1=>c_i,\ ?C2=>c_i)\}_{i=1..900}]$ (Joining this clause discards courses taught by other faculty.)
4	$[\{(?S1=>s_i,\ ?C1=>c_i,\ ?C2=>c_i)\}_{i=1..60}]$

The output from the backward chaining reasoner will be a query response. These must be merged into the currentSolutionSpace as a set of additional restrictions. Fig. 2 shows how this is done.

Each binding already in the SolutionSpace ❶ that shares at least one variable with the new binding ❷ is applied to the new binding, updating the new binding so that its domain is the union of the sets of variables in the old and new bindings and the specific functions represent the constrained cross-product (join) of the two. Any such old bindings so joined to the new one can then be discarded.

The join function at ❷ returns the joined QueryResponse as an update of its first parameter. The join operation is carried out as a hash join [23] with an average complexity $O(n_1+n_2+m)$ where the n_i are the number of tuples in the two input sets and m is the number of tuples in the joined output.

The third (boolean) parameter of the join call indicates whether the join is forced (true) or optional (false), and the boolean return value indicates whether an optional join was actually carried out. Our intent is to experiment in future versions with a dynamic decision to defer optional joins if a partial calculation of the join reveals that the output will far exceed the size of the inputs, in hopes that a later query clause may significantly restrict the tuples that need to participate in this join.

As noted earlier, our interpretation of the SolutionSpace is that it denotes a set of potential bindings to variables, represented as the join of an arbitrary number of QueryResponses. The actual computation of the join can be deferred, either because of a dynamic size-based criterion as just described, or because of the requirement at ❶ that joins be carried out immediately only if the input QueryResponses share at least one variable. In the absence of any such sharing, a join would always result in an output size as long as the products of its input sizes. Deferring such joins can help reduce the size of the SolutionSpace and, as a consequence, the

TABLE III. TRACE OF JOIN OF CLAUSES IN ASCENDING ORDER OF ESTIMATED SIZE

Clause Being Joined	Resulting SolutionSpace
(initial)	[ ]
3	$[[\{(?C1=>c_i)\}_{i=1..3}]$
4	$[\{(?C1=>c_i,\ ?C2=>c_j)\}_{i=1..3,\ j=1..3}]$
1	$[\{(?S1=>s_i,\ ?C1=>c_i,\ ?C2=>c'_i)\}_{i=1..270}]$
2	$[\{(?S1=>s_i,\ ?C1=>c_i,\ ?C2=>c_i)\}_{i=1..60}]$

```
void SolutionSpace::restrictTo (QueryRe-
sponsenewbinding)
{
   for each element oldBinding
     in solutionSpace
   {
      if (newbinding shares variables
          with oldbinding){❶
         bool merged = join(newBinding,
           oldBinding,false);❷
         if (merged) {
            remove oldBinding from
               solutionSpace;
         }
      }
   }
   add newBinding to solutionSpace;
}
```

Figure 2. Restricting a SolutionSpace.

cost of subsequent joins.

When all clauses of the original query have been processed (Fig. 1❻), we may have deferred several joins because they involved unrelated variables or because they appeared to lead to a combinatorial explosion on their first attempt. The finalJoin function shown in Fig.3 is tasked with reducing the internal SolutionSpace to a single QueryResponse, carrying out any join operations that were deferred by the earlier restrictTo calls. In many ways, finalJoin is a recap of the answerAQuery and restrictTo functions, with two important differences:

- Although we still employ a greedy ordering ❶to reduce the join sizes, there is no need for estimated sizes because the actual sizes of the input QueryResponses are known.
- There is no longer an option to defer joins between QueryResponses that share no variables. All joins must be performed in this final stage❷ and so the "forced" parameter to the optional join function is set to true.

For example, suppose that we were processing a different example query to determine which mathematics courses are taken by computer science majors, represented as the sequence of the following QueryPatterns, shown with their estimated sizes in Table IV.

```
QueryResponseSolutionSpace::finalJoin ()
{
   sort the bindings in this solution
     space into ascending order by
     number of tuples;  ❶

   QueryResponse result = first of the
     sorted bindings;
   for each remaining binding b
     in solutionSpace {
      join (result, b, true);  ❷
   }
   return result;
}
```

Figure 3. Final Join.

TABLE IV. EXAMPLE QUERY 2

Clause	QueryPattern	Query Response
1	(?S1 takesCourse ?C1)	$\{(?S1=>s_j,?C1=>c_j)\}_{j=1..100,000}$
2	(?S1 memberOf CSDept)	$\{(?S1=>s_j)\}_{j=1..1,000}$
3	(?C1 taughtby ?F1)	$\{(?C1=>c_j, ?F1=>f_j)\}_{j=1..1,500}$
4	(?F1 worksFor MathDept)	$\{(?F1=>f_i)\}_{i=1..50}$

To illustrate the effect of deferring joins on responses that do not share variables, even with the greedy ordering discussed earlier, suppose, first, that we perform all joins immediately. Assuming the greedy ordering that we have already advocated, the trace of the answerAQuery algorithm is shown in Table V.

In the prototype from which this example is taken, the Math department teaches 150 different courses and there are 1,000 students in the CS Dept. Consequently, the merge of clause 3 (1,500 tuples) with the SolutionSpace then containing 50,000 tuples yields considerably fewer tuples than the product of the two input sizes. The worst step in this trace is the final join, between sets of size 100,000 and 150,000.

But consider that the join of clause 2 in that trace was between sets that shared no variables. If we defer such joins, then the first SolutionSpace would be retained "as is". The resulting trace is shown in Table VI.

The subsequent addition of clause 3 results in an immediate join with only one of the responses in the solution space. The response involving ?S1 remains deferred, as it shares no variables with the remaining clauses in the SolutionSpace. The worst join performed would have been between sets of size 100,000 and 150, a considerable improvement over the non-deferred case.

IV. EVALUATION OF QUERY OPTIMIZATION

In this section, we compare our answerAQuery algorithm of Fig. 1 against an existing system, Jena, that also answers queries via a combination of an in-memory backward chaining reasoner with basic knowledge base retrievals.

The comparison was carried out using two LUBM benchmarks consisting of one knowledge base describing a single university and another describing 10 universities. Prior to the application of any reasoning, these benchmarks contained 100,839 and 1,272,871 triples, respectively.

We evaluated these using a set of 14 queries taken from LUBM [22]. These queries involve properties associated with the LUBM university-world ontology, with none of the custom properties/rules whose support is actually our end

TABLE V. TRACE OF JOIN OF CLAUSES IN ASCENDING ORDER OF ESTIMATED SIZE

Clause Being Joined	Resulting SolutionSpace
(initial)	[]
4	$[\{(?F1=>f_i)\}_{i=1..50}]$
2	$[\{(?F1=>f_i, ?S1=>s_i)\}_{i=1..50,000}]$
3	$[\{(?F1=>f_i, ?S_1=>s_i, ?C1=>c_i)\}_{i=1..150,000}]$
1	$[\{(?F1=>f_i, ?S1=>s_i, ?C1=>c_i)\}_{i=1..1,000}]$

TABLE VI. TRACE OF JOIN OF CLAUSES WITH DEFERRED JOINS

Clause Being Joined	Resulting SolutionSpace
(initial)	[]
4	$[\{(?F1=>f_i)\}_{i=1..50}]$
2	$[\{(?F1=>f_i)\}_{i=1..50},\{(?S1=>s_j)\}_{j=1..1,000}]$
3	$[\{(?F1=>f_i, ?C1=>c_i)\}_{i=1..150}, \{(?S1=>s_j)\}_{j=1..1,000}]$
1	$[\{(?F1=>f_i, ?S1=>s_i, ?C1=>c_i)\}_{i=1..1,000}]$

goal (as discussed in [3]). Answering these queries requires, in general, reasoning over rules associated with both RDFS and OWL semantics, though some queries can be answered purely on the basis of the RDFS rules.

Table VII compares our algorithm to the Jena system using a pure backward chaining reasoner. Our comparison focuses on response time, as our optimization algorithm should be neutral with respect to result accuracy, offering no more and no less accuracy than is provided by the interposed reasoner.

As a practical matter, however, Jena's system cannot process all of the rules in the OWL semantics rule set, and was therefore run with a simpler ruleset describing only the RDFS semantics. This discrepancy accounts for the differences in result size (# of tuples) for several queries. Result sizes in the table are expressed as the number of tuples returned by the query and response times are given in seconds. An entry of "n/a" means that the query processing had not completed (after 1 hour).

Despite employing the larger and more complicated rule set, our algorithm generally ran faster than Jena, sometimes by multiple orders of magnitude. The exceptions to this trend are limited to queries with very small result set sizes or queries 10-13, which rely upon OWL semantics and so could not be answered correctly by Jena. In two queries (2 and 9), Jena timed out.

Jena also has a hybrid mode that combines backward chaining with some forward-style materialization. Table VIII shows a comparison of our algorithm with a pure backward chaining reasoner against the Jena hybrid mode. Again, an "n/a" entry indicates that the query processing had not completed within an hour, except in one case (query 8 in the 10 Universities benchmark) in which Jena failed due to exhausted memory space.

The times here tend to be someone closer, but the Jena system has even more difficulties returning any answer at all when working with the larger benchmark. Given that the difference between this and the prior table is that, in this case, some rules have already been materialized by Jena to yield, presumably, longer lists of tuples, steps taken to avoid possible combinatorial explosion in the resulting joins would be increasingly critical.

V. OPTIMIZED BACKWARD CHAINING ALGORITHM

When the knowledge base is dynamic, backward chaining is a suitable choice for ontology reasoning. However, as the size of the knowledge base increases, standard backward chaining strategies [2][15] do not scale well for ontology reasoning. In this section, first, we discuss issues some backward chaining methods expose for ontology reasoning. Second, we present our backward chaining algorithm that introduces new optimization techniques as well as addresses the known issues.

A. Issues

1. Guaranteed Termination: Backward chaining is usually implemented by employing a depth-first search strategy. Unless methods are used to prevent it, the depth-first search could go into an infinite loop. For example, in our rule set, we have rules that involve each other when proving their heads:

rule1: (?P owl:inverseOf ?Q) -> (?Q owl:inverseOf ?P)
rule2;(?P owl:inverseOf ?Q), (?X ?P ?Y) -> (?Y ?Q ?X)

TABLE VII COMPARISON AGAINST JENA WITH BACKWARD CHAINING

LUBM:	1 University, 100,839 triples				10 Universities, 1,272,871 triples			
	answerAQuery		*Jena Backwd*		*answerAQuery*		*Jena Backwd*	
	response time	*result size*	*response time*	*result size*	*response time*	*result size*	*response time*	*result size*
Query1	0.20	4	0.32	4	0.43	4	0.86	4
Query2	0.50	0	130	0	2.1	28	n/a	n/a
Query3	0.026	6	0.038	6	0.031	6	1.5	6
Query4	0.52	34	0.021	34	1.1	34	0.41	34
Query5	0.098	719	0.19	678	0.042	719	1.0	678
Query6	0.43	7,790	0.49	6,463	1.9	99,566	3.2	82,507
Query7	0.29	67	45	61	2.2	67	8,100	61
Query8	0.77	7,790	0.91	6,463	3.7	7,790	52	6,463
Query9	0.36	208	n/a	n/a	2.5	2,540	n/a	n/a
Query10	0.18	4	0.54	0	1.8	4	1.4	0
Query11	0.24	224	0.011	0	0.18	224	0.032	0
Query12	0.23	15	0.0020	0	0.33	15	0.016	0
Query13	0.025	1	0.37	0	0.21	33	0.89	0
Query14	0.024	5,916	0.58	5,916	0.18	75,547	2.6	75,547

TABLE VIII. COMPARISON AGAINST JENA WITH WITH HYBRID REASONER

LUBM	1 University, 100,839 triples				10 Universities, 1,272,871 triples			
	answerAQuery		*Jena Hybrid*		*answerAQuery*		*Jena Hybrid*	
	response time	*result size*	*response time*	*result size*	*response time*	*result size*	*response time*	*result size*
Query1	0.20	4	0.37	4	0.43	4	0.93	4
Query2	0.50	0	1,400	0	2.1	28	n/a	n/a
Query3	0.026	6	0.050	6	0.031	6	1.5	6
Query4	0.52	34	0.025	34	1.1	34	0.55	34
Query5	0.098	719	0.029	719	0.042	719	2.7	719
Query6	0.43	7,790	0.43	6,463	1.9	99,566	3.7	82,507
Query7	0.29	67	38	61	2.2	67	n/a	n/a
Query8	0.77	7,790	2.3	6,463	3.7	7,790	n/a	n/a
Query9	0.36	208	n/a	n/a	2.5	2,540	n/a	n/a
Query10	0.18	4	0.62	0	1.8	4	1.6	0
Query11	0.24	224	0.0010	0	0.18	224	0.08	0
Query12	0.23	15	0.0010	0	0.33	15	0.016	0
Query13	0.025	1	0.62	0	0.21	33	1.2	0
Query14	0.024	5,916	0.72	5,916	0.18	75,547	2.5	75,547

In order to prove body clause ?P owl:inverseOf ?Q in rule1, we need to prove the body of rule2 first, because the head of rule2 matches body clause ?P owl:inverseOf ?Q. In order to prove the first body clause ?P owl:inverseOf ?Q in rule2, we also need to prove the body clause ?P owl: inverseOf ?Q in rule1, because the head of rule1 matches body clause ?P owl:inverseOf ?Q.

Even in cases where depth-first search terminates, the performance may suffer due to time spent exploring, in depth, branches that ultimately do not lead to a proof.

We shall use the OLDT [24] method to avoid infinite recursion and will introduce optimizations aimed at further performance improvement in Section VI.C.

2. The owl:sameAs Problem: The built-in OWL property owl:sameAs links two equivalent individuals. An owl:sameAs triple indicates that two linked individuals have the same "identity" [25]. An example of a rule in the OWL-Horst rule set that involves the owl:sameAs relations is the rule: "(?x owl:sameAs ?y) (?x ?p ?z) -> (?y ?p ?z)".

Consider a triple, which has m owl:sameAs equivalents of its subject, n owl:sameAs equivalents of its predicate, and k owl:sameAs equivalents of its object, Then m*n*k triples would be derivable from that triple.

Reasoning with the owl:sameAs relation can result in a multiplication of the number of instances of variables during backward-chaining and expanded patterns in the result. As long as that triple is in the result set, all of its equivalents would be in the result set as well. This adds cost to the reasoning process in both time and space.

B. The Algorithm

The purpose of this algorithm is to generate a query response for a given query pattern based on a specific rule set. We shall use the following terminology.

A VariableBinding is a substitution of values for a set of variables.

A RuleSet is a set of rules for interpretation by the reasoning system. This can include RDFS Rules [26], Horst rules [27] and custom rules [28] that are used for ontology reasoning. For example,

[rdfs1: (?x ?p ?y) -> (?p rdf:type rdf:Property)].

The main algorithm calls the function BackwardChaining, which finds a set of triples that can be unified with pattern with bindings varList, any bindings to variables appearing in headClause from the head of applied rule, bodylist that are reserved for solving the recursive problem. Given a Goal and corresponding matched triples, a QueryResponse is created and returned in the end.

Our optimized BackwardChaining algorithm, described in Fig. 4, is based on conventional backward chaining algorithms [2]. The solutionList is a partial list of solutions already found for a goal.

For a goal that has already been resolved, we simply get the results from solutionList. For a goal that has not been resolved yet, we will seek a resolution by applying the rules. We initially search in the knowledge base to find triples that match the goal (triples in which the subject, predicate and object are compatible with the query pattern). Then, we find rules with heads that match the input pattern. For each such rule we attempt to prove it by proving the body clauses (new goals) subject to bindings from already-resolved goals from the same body. The process of proving one rule is explained below. The method of "OLDT" [24] is adopted to solve the non-termination issue we mentioned in Section VI.C. Finally, we apply any "same as" relations to candidateTriples to solve the owl:sameAs problem. During this process of "SameAsTripleSearch", we add all equivalent triples to the existing results to produce complete results.

Fig. 5 shows how to prove one rule, which is a step in Fig. 4. The heart of the algorithm is the loop through the clauses of a rule body, attempting to prove each clause. Some form of selection function is implied that selects the next unproven clause for consideration on each iteration. Traditionally, this would be left-to-right as the clauses are written in the rule. Instead, we order the body clauses by the number of free variables. The rationale for this ordering will be discussed in the following Section VI. A.

```
BackwardChaining(pattern,headClause,bodylist,level,varList)
{
   if (pattern not in solutionList){
      candidateTriples+= matches to pattern that found in knowledge base;
      solutionList+= mapping from pattern to candidateTriples;
      relatedRules = rules with matching heads to pattern that found in ruleList;
      realizedRules = all the rules in relatedRules with substitute variables from pattern;
      backupvarList = back up clone of varList;
      for (each oneRule in realizedRules){
         if(attemptToProveRule(oneRule, varList, level)){
            resultList= unify(headClause, varList);
            candidateTriples+= resultList;
         }
         oldCandidateTriples = triples in mappings to headClause from solutionList;
         if ( oldCandidateTriples not contain candidateTriples){
            update solutionList with candidateTriples;
            if(UpdateafterUnificationofHead(headClause, resultList))
            {
                newCandidateTriples = triples in mappings to headClause from solutionList;
                candidateTriples+= newCandidateTriples;
            }
         }
       }
    }
    else /* if (solutionList.contains(pattern)) */
    {
       candidateTriples+= triples in mappings to pattern from solutionList;
       Add reasoning context, including head and bodyRest to lookupList;
    }
    SameAsTripleSearch(candidateTriples);
    return candidateTriples;
}
```

Figure 4. Process of BackwardChaining.

The process of proving one goal (a body clause from a rule) is given in Fig. 6. Before we prove the body clauses (new goals) in each rule, the value of a calculated dynamic threshold decides whether we perform the substitution or not. We substitute the free variables in the body clause with bindings from previously resolved goals from the same body. The step helps to improve the reasoning efficiency in terms of response time and scalability and will be discussed in Section VI.B. We call the BackwardChaining function to find a set of triples that can be unified with body clause (new goal) with substituted variables. Bindings will also be updated gradually following the proof of body clauses.

```
attemptToProveRule(oneRule,varList,level)
{
   body = rule body of oneRule;
   sort body by ascending number of free
      variables;
   head = rule head of oneRule;
   for (each bodyClause in body)
   {
      canBeProven =
        attemptToProveBodyClause (
          bodyClause, body, head,
          varList, level);
       if (!canBeProven) break;
   }
   return canBeProven;
}
```

Figure 5. Process of proving one rule.

VI. OPTIMIZATION DETAILS & DISCUSSION

There are four optimizations that have been introduced in our algorithm for backward chaining. These optimizations are: 1) the implementation of the selection function, which implements the ordering the body clauses in one rule by the number of free variables, 2) the upgraded substitute function, which implements the substitution of the free variables in the body clauses in one rule based on calculating a threshold that switches resolution methods, 3) the application of OLDT and 4) solving of the owl:sameAs problem. Of these, optimization 1 is an adaptation of techniques employed in other reasoning contexts [29][30] and optimizations 3 and 4 have appeared in [24, 31] whereas techniques 2 are new. We will describe the implementation details of these optimizations below. A preliminary evaluation of these techniques is reported in a separate paper. [3] A more extensive evaluation is reported here in Section VII.

A. Ordered Selection Function

The body of a rule consists of a conjunction of multiple clauses. Traditional SLD (Selective Linear Definite) clause resolution systems such as Prolog would normally attempt these in left-to-right order, but, logically, we are free to attempt them in any order.

```
attemptToProveBodyClause(goal, body,
head, varList, level)
{
  canBeProven = true;
  dthreshold = Calculate dynamic
    threshold;
  patternList = get unified patterns by
    replacing variables in bodyClause
    from varList for current level with
    calculated dthreshold;
  for(each unifiedPattern in
      patternList ) {
     if(!unifiedPattern.isGround()) {
         bodyRest = unprocessedPartOf(
           body, goal);
        triplesFromResolution+=
           BackwardChaining(
             unifiedPattern, head,
             bodyRest, level+1,
             varList);
     }
     else if(unifiedPattern.isGround()) {
        if (knowledgeBase contains
             unifiedPattern){
           triplesFromResolution+=
             unifiedPattern;
        }
     }
  }
  if(triplesFromResolution.size()>0) {
     update_varList with varList,
       triplesFromResolution, goal, and
       level;
     if (varList==null) {
        canBeProven = false;
     }
  }
  else{
      canBeProven = false;
  }
  return canBeProven;
}
```

Figure 6. Process of proving one goal.

We expect that given a rule under proof, ordering the body clauses into ascending order by the number of free variables will help to decrease the reasoning time. For example, let us resolve the goal "?y rdf:type Student", and consider the rule:

[rdfs3: (?x ?p ?y) (?p rdfs:range ?c) -> (?y rdf:type ?c)]

The goal "?y rdf:type Student" matches the head of rule "?y rdf:type ?c", and ?c is unified with Student.

If we select body clause "?x ?p ?y" to prove first, it will yield more than 5 million (using LUBM(40) [22]) instances of clauses. The proof of body clause "?x ?p ?y" in backward chaining would take up to hours. Result bindings of "?p" will be propagated to the next body clause "?p rdfs:range ?c" to yield new clauses (p1 rdfs:range Student), (p2 rdfs:range Student), …, (p32 rdfs:range Student), and then a separate proof would be attempted for each of these specialized forms.

If we select body clause "?p rdfs:range Student" (?c is unified with Student) to prove first, it will yield zero (using LUBM(40)) instances of clauses. The proof of body clause "?p rdfs:range Student" would take up to seconds. No result bindings would be propagated to body clause "?x ?p ?y". The process of proof terminates.

The body clause "?p rdfs:range ?c" has one free variable ?p while the body clause "?x ?p ?y" has three free variables. It is reasonable to prove body clause with fewer free variables first, and then propagate the result bindings to ?p to next body clause "?x ?p ?y". Mostly, goals with fewer free variables cost less time to be resolved than goals with more free variables, since fewer free variables means more bindings and body clauses with fewer free variables will match fewer triples.

B. *Switching between Binding Propagation and Free Variable Resolution*

Binding propagation and free variable resolution are two modes of for dealing with conjunctions of multiple goals. We claim that dynamic selection of these two modes during the reasoning process will increase the efficiency in terms of response time and scalability.

These modes differ in how they handle shared variables in successive clauses encountered while attempting to prove the body of a rule. Suppose that we have a rule body containing clauses (?x p1 ?y) and (?y p2 ?z) [other patterns of common variables are, of course, also possible] and that we have already proven that the first clause can be satisfied using value pairs $\{(x_1, y_1), (x_2,y_2),\ldots(x_n,y_n)\}$.

In the binding propagation mode, the bindings from the earlier solutions are substituted into the upcoming clause to yield multiple instances of that clause as goals for subsequent proof. In the example given above, the value pairs from the proof of the first clause would be applied to the second clause to yield new clauses (y1 p2 ?z), (y2 p2 ?z), …, (yn p2 ?z), and then a separate proof would be attempted for each of these specialized forms. Any (y,z) pairs obtained from these proofs would then be joined to the (x,y) pairs from the first clause.

In the free variable resolution mode, a single proof is attempted of the upcoming clause in its original form, with no restriction upon the free variables in that clause. In the example above, a single proof would be attempted of (?y p2 ?z), yielding a set of pairs $\{(y_n, z_1), (y_{n+1},z_2),\ldots(x_{n+k},z_k)\}$. The join of this with the set $\{(x_1, y_1), (x_2,y_2),\ldots(x_n,y_n)\}$ would then be computed to describe the common solution of both body clauses.

The binding propagation mode is used for most backward chaining systems [15]. There is a direct tradeoff of multiple proofs of narrower goals in binding propagation against a single proof of a more general goal in free variable resolution. As the number of tuples that solve the first body clause grows, the number of new specialized forms of the subsequent clauses will grow, leading to higher time and space cost overall. If the number of tuples from the earlier clauses is large enough, free variable resolution mode will be more efficient. (In the experimental results in Section VII, we will demonstrate that neither mode is uniformly faster across all problems.)

Following is an example (using LUBM(40)) showing one common way of handling shared variables between body clauses.

Suppose we have an earlier body clause 1: "?y type Course" and a subsequent body clause 2: "?x takesCourse ?y". These two clauses have the common variable ?y. In our experiments, it took 1.749 seconds to prove body clause 1 while it took an average of 0.235 seconds to prove body clause 2 for a given value of ?y from the proof of body clause 1. However, there were 86,361 students satisfying variable ?x, which means it would take 0.235 *86,361=20,295 seconds to finish proof of 86,361 new clauses after applying value pairs from the proof of body clause 1. 20,295 seconds is not acceptable as query response time. We need to address this problem to improve reasoning efficiency in terms of response time and scalability.

We propose to dynamically switch between modes based upon the size of the partial solutions obtained so far. Let n denote the number of solutions that satisfy an already proven clause. Let t denote threshold used to dynamically select between modes. If $n \leqslant t$, then the binding propagation mode will be selected. If n>t, then the free variable resolution mode will be selected. The larger the threshold is, the more likely binding propagation mode will be selected.

Suppose that we have a rule body containing clauses (a1 p1 b1) (a2 p2 b2). Let (a1 p1 b1) be the first clause, and (a2 b2 c2) be the second clause. ai, bi and ci ($i \in [1,2]$) could be free variable or concrete value. Assume that there is at least one common variable between two clauses.

In the binding propagation mode, the value pairs from the proof of the first clause would be applied to the second clause to yield new clauses ($a2_1$ $p2_1$ $b2_1$), ($a2_2$ $p2_2$ $b2_2$), …, ($a2_n$ $p2_n$ $c2_n$), and then a separate proof would be attempted for each of these specialized forms. Any value sets obtained from these proofs would then be joined to the value sets from the first clause. Let $join_1$ denote the time spent on the joint operations. Let $proof_1^i$ denote the time of proving first clause with i free variables and $proof_2^j$ be the average time of proving new specialized form with j free variables. ($i \in [1,3]$, $j \in [0,2]$)

In the free variable resolution mode, a single proof is attempted of the upcoming clause in its original form, with no restriction upon the free variables in that clause. A single proof would be attempted of (a2 p2 b2), yielding a set of value sets. The join of the value sets yielded from the first clause and the values sets yielded from the second clause would then be computed to describe the common solution of both body clauses. Let $join_2$ denote the time spent on the joint operations. Let $proof_3^k$ denote the time of proving second clause with k free variables. ($k \in [1,3]$)

Determining t is critical to switching between two modes. Let us compare the time spent on binding propagation mode and free variable resolution mode to determine t. Binding propagation is favored when

$$proof_1^i + proof_2^j * n + join_1 < proof_1^i + proof_3^k + join_2$$

Isolating the term involving n,

$$proof_2^j *n < proof_1^i + proof_3^k + join_2 - proof_1^i - join_1$$

$$proof_2^j *n < proof_3^k + join_2 - join_1$$

$join_1$ is less than or equal to $join_2$, because the value sets from the second clause in the binding propagation mode have already been filtered by the value sets from the first clause first. The join operations in binding propagation mode are therefore a subset of the join operations in free variable resolution mode. Let t be the largest integer value such that

$$proof_2^j *t < proof_3^k$$

then

$$proof_2^j *t <= proof_2^j *n < proof_3^k + join_2 - join_1$$

We conclude that:

$$t = \text{floor}(proof_3^k / proof_2^j) \qquad (1)$$

Formula (1) provides thus a method for calculating the threshold t that determines when to employ binding propagation. In that formula, k denotes the number of free variables in the second clause (a2 p2 b2), j denotes the number of free variables of the new specialized forms ($a2_1$ $p2_1$ $b2_1$), ($a2_2$ $p2_2$ $b2_2$), ($a2_n$ $p2_n$ $c2_n$) of the second clause with ($k \in [1,3]$, $j \in [0,2]$). The specialized form of the second clause has one or two less free variables than the original form. Hence, the possible combinations of (k,j) are {(3,2), (3,1), (2,1), (2,0), (1,0)}.

To estimate $proof_3^k$ and $proof_2^j$, we record the time spent on proving goals with different numbers of free variables. We separately keep a record of the number of goals that have one free variable, two free variables and three free variables after we start calling our optimized backwardChaining algorithm. We also record the time spent on proving these goals. After we have recorded a sufficient number of proof times (experiments will give us an insight into what constitutes a 'sufficient' number), we compute the average time spent on goals with k free variables and j free variables, respectively, to obtain an estimate of $proof_3^k$ and proof2j.

In order to adopt accurate threshold to help improve the efficiency, we apply different thresholds to different situations with corresponding number of free variable set (k,j).

We assign the initial value to t from previous experiments in a particular knowledge base/query environment if they exist or zero otherwise.

We update the threshold several times when answering a particular query. The threshold will change as different queries are being answered. For each query, we will call the optimized backward chaining algorithm recursively several times. Each call of backwardChaining is given a specific goal as an input. During the running of backwardChaining, the average time of proving a goal as a function of the number of free variables will be updated after a goal has been proven. During the running of backwardChaining, every time before making selection between two modes the estimate threshold is updated before making the decision.

C. *How to Avoid Repetition and Non-Termination*

Given RDFS Rules [26], Horst rules [27] and custom rules [28] in the rule set and queries for answering, backward chaining for ontology reasoning may hit the same goals for several times. Some body clauses such as ?a rdfs:subClassOf ?b and ?x rdfs:subPropertyOf ?y appear in multiple rules in Horst rule set that is used in many reasoning systems. During the process of answering a given query, these rules containing the same body clauses might be necessary to be proved to answer the query. During the process of answering a given query, some rules may be repeatedly called for more than one time, leading to proving the same body clause like ?a rdfs:subClassOf ?b more than one time. Within the process of answering one query, such a repetition decreases the efficiency in terms of response time. Backward chaining with memorization will help to avoid repetition.

Backward chaining is implemented in logic programming [32] by SLD resolution [33]. When we apply conventional backward chaining process to ontology reasoning, it has the same non-termination problem as SLD resolution does. During the proving process, the rule body needs to be satisfied to prove the goal. In some cases, the rule body requires proving goals that have the same property as the goal, resulting possibly in an infinite loop unless steps are taken to ensure termination.

For example, [rdfs8: (?a rdfs:subClassOf ?b), (?b rdfs:subClassOf ?c) -> (?a rdfs:subClassOf ?c)] is one rule in the RDFS rule set used for ontology reasoning. When we apply standard backward chaining to ontology reasoning, proving the head (?a rdfs:subClassOf ?c) requires proving of the body (?a rdfs:subClassOf ?b) and (?b rdfs:subClassOf ?c). This loop will be infinite without applying any techniques.

We use an adaptation of the OLDT algorithm to solve this non-termination problem. The OLDT algorithm is an extension of the SLD-resolution [33] with a left to right computation rule. OLDT maintains a solution table and lookup table to solve the recursion problem.

D. owl:sameAs Optimization

The "owl:sameAs" relation poses a problem [31] for almost all the reasoning systems including forward chaining. In our reasoning system, we first pre-compute all possible owl:sameAs pairs and save them to a sameAs table. Second, we select a representative node to represent an equivalence class of owl:sameAs URIs. Third, we replace the equivalence class of owl:sameAs URIs with the representative node. At last, if users want to return all the identical results, we populate the query response using the sameAs table by replacing the representative node with the URIs in the equivalence class.

As we described in Section V, reasoning with the owl:sameAs relation can result in a multiplication of the number of instances of variables during backward-chaining and expanded patterns in the result. As long as that triple is in the result set, all of the members in its equivalence class would be in the result set as well. This adds cost to the reasoning process in both time and space. The optimization that applies pre-computation and selects a representative node improves the performance in terms of time and space.

This optimization is a novel adaptation of owl:sameAs optimization in forward chaining reasoning system, such as OWLIM-SE [34] and Oracle [13], to backward chaining reasoning systems.

VII. BACKWARD CHAINING WITH EXTERNALLY STORED KNOWLEDGE BASE

In Section IV and in our earlier experiments assessing the effectiveness of our optimized reasoner [3], all our experiments were performed 'in-memory', which limited the study to a knowledge base of less than 10 Million triples.

In this section, we switch to implementations that use external storage for the knowledge base. We consider Jena SDB [35], Jena TDB [36] and OWLIM-SE [34]. We extend our study based on a knowledge base of more than 100 Million triples.

The employment of external storage introduces new factors and has implications on how to improve the scalability of our backward chaining reasoner. First, any optimization technique needs to balance the number of accesses to data and the size of the retrieved data against the size of in-memory cache and its use. Second, the algorithm has to take now into account that it will take longer to access a triple (or a set of triples) due to having to perform I/O. In-memory reasoners typically have a 'model' of the knowledge base in which they store the facts and an API to access them. When an external storage is used they would provide transparent connections from the model to the external databases that would allow the reasoner to use the same API for accessing the model. This leads to a third factor effecting the scalability and performance of the reasoner: the middleware that realizes the transparent linking.

Jena SDB provides persistent triple stores using relational databases. An SQL database is required for the storage and query of triples for SDB. In this paper, we used MySQL and PostgreSQL as the relational database for SDB. Jena TDB is claimed as a more scalable and faster triple store than SDB [35].A special Jena adapter permits access to OWLIM-SE repositories [34]. Reasoners can access all three storage systems via a common Jena API.

A. Preliminary Analysis

We begin by exploring the relative impact on overall performance of the three major components of the backward chaining reasoner, the middleware, and the storage system itself. The purpose of this analysis is to determine how much time we can save by improving any one of these subsystems in isolation.

We employed Jena SDB + MySQL as the external storage for our backward chaining reasoner in the experiment, evaluating the query response time of 14 queries from LUMB [22] using LUBM(30).

A single function in our backward chaining algorithm implementation is responsible for all data retrievals from the triple store. We refer to this function as "the Data-retrieval function" in the remainder of this section. Data-retrieval function in this paper. We recorded the clock time T_f and CPU time t_f spent within the Data-retrieval function and in the whole query processing (T_{tot} and t_{tot}, respectively) in Table IX.

The portion of the CPU and clock times spent I answering the query but not spent in the Data-retrieval function is attributable to the backward chaining reasoner:

TABLE IX CLOCK TIME, CPU TIME AND I/O TIME FROM EXPERIMENTS WITH JENA SDB USING LUBM (30)

	Total Clock time, T_{tot}	Total CPU Time, t_{tot}	Clock time in Data-retrieval function, T_f	CPU time in I/O function, t_f
Query1	1405.00	951.00	920.00	546.00
Query2	9631.00	6084.00	5058.00	2293.00
Query3	203.00	78.00	109.00	31.00
Query4	35354.00	8096.00	31140.00	5070.00
Query5	173.00	78.00	94.00	15.00
Query6	23744.00	7035.00	19984.00	3712.00
Query7	24058.00	9984.00	18659.00	6333.00
Query8	28694.00	11029.00	22680.00	5896.00
Query9	29598.00	11700.00	23899.00	6988.00
Query 10	18612.00	6630.00	15040.00	3572.00
Query 11	3636.00	561.00	2964.00	124.00
Query 12	7567.00	1903.00	5226.00	405.00
Query 13	187.00	46.00	95.00	0.00
Query 14	1873.00	811.00	1451.00	452.00

$T_{bw} = T_{tot} – T_f$
$T_{bw} = t_{tot} – t_f$

The clock time observed during the Data-retrieval function includes actual input operations on the underlying triple store, together with the CPU-intensive manipulation of the input data by the middleware layer. Assuming that the ratio, $\rho = t_{tot}/T_{tot}$, of CPU time to clock time observed over the processing of an entire query would remain approximately constant during the middleware CPU, we were able to estimate the portion of the Data-retrieval function clock time that was attributable to the middleware:

$T_{mid} = \rho\ t_{mid}$

and can attribute the remaining clock time as the actual time spent doing I/O:

$T_{IO} = T_f – T_{mid}$

Then we can estimate a minimal clock time to answer the query, assuming 100% CPU utilization, as

$T_{min} = t_{bw} + \rho\ T_{mid} + T_{IO}$

Table X shows the values of these estimates, together the percentage of that value attributable to each of the three components. In Table X, the percentage of time spent in I/O operations ranges from 22% to 75%, a considerable variation. This might be because some retrievals from triple store retrieve huge numbers of triples while others are far more focused and process much less data.

TABLE X ESTIMATED I/O TIME AND IDEAL PERCENTAGES FROM EXPERIMENTS WITH JENA SDB USING LUBM (30)

	Min possible clock time to answer a query, T_{min}	% of T_{min} spent in I/O	% of Tmin spent in BW chaining	% of T_{min} time spent in middleware
Query1	1217.15	0.22	0.33	0.45
Query2	8376.00	0.27	0.45	0.27
Query3	125.00	0.38	0.38	0.25
Query4	32175.53	0.75	0.09	0.16
Query5	153.19	0.49	0.41	0.10
Query6	22818.84	0.69	0.15	0.16
Query7	19277.93	0.48	0.19	0.33
Query8	26801.04	0.59	0.19	0.22
Query9	27147.26	0.57	0.17	0.26
Query 10	17497.60	0.62	0.17	0.20
Query 11	3334.32	0.83	0.13	0.04
Query 12	6496.09	0.71	0.23	0.06
Query 13	141.00	0.67	0.33	0.00
Query 14	1730.68	0.53	0.21	0.26

The percentage of the time devoted to the middleware ranges from 0% to 44%, with an average around 20%, indicating that the triple storage layer adds a significant component of CPU time. Our backward chaining code running on top of that accounts for 13 to 45% of minimal processing time, and the average is 25%.

These percentages are surprisingly balanced, suggesting that improvements to any one of the three major components of the system can have only modest effect on the total time. Dramatic improvements will be possible only by improvement in all three areas. One possible avenue of exploration is changes to the reasoner that would not only speed up the reasoner but would affect the number and size of requests for input from the underlying store. Indirectly, at least, several of the optimizations we have proposed in Section VI could have such an effect. Caching, an effect not explored in this experiment, could also have a major impact across all three areas.

B. *Evaluation of the Optimization techniques*

In this section, we examine the impact of the two major optimizations proposed in Section VI.

1) Ordered Selection Function

We have proposed replacing the traditional left-to-right processing of clauses within rule bodies by an ordering by ascending number of free variables.

Table XI compares our backward chaining algorithm with our clause selection based on free variable count to the traditional left-to-right selection on a relatively small knowledge base (100,839 triples) LUBM(1) [22] stored in Jena TDB. Traditional left-to-right selection has been used in Jena [15] and Prolog [32]. Backward chaining with the ordered selection function yields considerably smaller query response times for all the queries than left-to-right. The I/O time of accessing the external triple storage magnifies the problem of left-to-right selection compared to [3] because the knowledge base is in external triple storage TDB now.

The difference becomes even more dramatic for a larger knowledge base (1,272,871 triples), LUBM(10) stored in Jena TDB, as shown in Table XII. With left-to-right selection, we are unable to answer any query within 30 minutes, and out-of-memory errors occur for almost half of the queries. The I/O time of accessing the external triple storage magnifies the problem of left-to-right selection compared to [3] because the knowledge base is in external triple storage TDB now.

2) Switching between Binding Propagation and Free Variable Resolution

Binding propagation and free variable resolution are two modes of for dealing with conjunctions of multiple goals. We have proposed dynamic selection of these two modes during the reasoning process to increase the efficiency in terms of response time and scalability.

We compare our backward chaining algorithm with three different modes of resolving goals on LUBM(10) stored in Jena TDB in Table XIII. The first mode uses dynamic selection between binding propagation mode and free variable resolution mode. The second mode uses binding propagation mode only. The third mode uses free variable resolution mode only.

Table XIII shows that neither binding propagation mode nor free variable resolution mode is uniformly better than the other on all cases. From query 1 to query 5 and query 13, dynamic mode performs almost same as binding propagation mode. From query 6 to query 10, dynamic mode performs dramatically better than binding propagation mode with much less query response time. For query 11, query 12 and query 14, dynamic mode performs better than binding propagation mode with less query response time.

For query1, query3 and query 14 only, dynamic mode performs almost same as free variable resolution mode. For the other queries, dynamic mode performs dramatically better than free variable resolution mode with much less query response time. The query response times of query6 to query10 are less by orders of magnitude when running our algorithm with the dynamic selection mode in comparison compared to running with binding propagation mode only and free variable resolution mode only. In all cases the optimized version finishes faster than the better of the other two versions. Overall, the results in Table XIII confirm the advantage of dynamically selecting between propagation modes. The I/O time of accessing the external triple storage magnifies the problem of binding propagation mode only and free variable resolution mode only compared to [3] be-

TABLE XI. EVALUATION OF CLAUSE SELECTION OPTIMIZATION ON LUBM(1) USING TDB AS EXTERNAL STORAGE

	Time (ms), Ordered	**Time (ms), Left-to right**	**Result Size (triples)**
Query1	296	>6.0*105	4
Query2	811	>6.0*105	0
Query3	46	>6.0*105	6
Query4	1419	>6.0*105	34
Query5	31	>6.0*105	719
Query6	265	>6.0*105	7,790
Query7	234	>6.0*105	67
Query8	483	>6.0*105	7,790
Query9	202	>6.0*105	208
Query10	156	>6.0*105	4
Query11	218	>6.0*105	224
Query12	202	>6.0*105	15
Query13	15	>6.0*105	1
Query14	31	>6.0*105	5,916

TABLE XII. EVALUATION OF CLAUSE SELECTION OPTIMIZATION ON LUBM(10) USING TDB AS EXTERNAL STORAGE

	Time (ms), Ordered	**Time (ms), Left-to right**	**Result Size (triples)**
Query1	1045	OutOfMemoryError: Java heap space	4
Query2	2433	>2.0*106	28
Query3	31	>2.0*106	6
Query4	3744	>2.0*106	34
Query5	15	>2.0*106	719
Query6	1435	OutOfMemoryError	99,566
Query7	1903	OutOfMemoryError	67
Query8	2106	OutOfMemoryError	7,790
Query9	1918	OutOfMemoryError	2,540
Query10	1138	OutOfMemoryError	4
Query11	140	>2.0*106	224
Query12	358	>2.0*106	15
Query13	15	>2.0*106	33
Query14	187	>2.0*106	75,547

TABLE XIII . EVALUATION OF DYNAMIC SELECTION VERSUS BINDING PROPAGATION AND FREE VARIABLE MODES ON LUBM(10) USING TDB AS EXTERNAL STORAGE

	Time (ms), Dynamic selection	**Time (ms), Binding propagation only**	**Time (ms), Free variable resolution only**
Query1	1045	904	904
Query2	2433	2683	26535
Query3	31	15	15
Query4	3744	4149	41605
Query5	15	15	2244810
Query6	1435	>6.0*105	20514
Query7	1903	>6.0*105	20763
Query8	2106	>6.0*105	42831
Query9	1918	>6.0*105	21512
Query10	1138	>6.0*105	19921
Query11	140	904	19094
Query12	358	1435	41745
Query13	15	31	24117
Query14	187	1154	187

cause the knowledge base are in external triple storage TDB now. The selection of the threshold in dynamic mode would be affected by the employment of external storage and affect the number of accesses to store.

C. Storage System Impact

To explore the effect of switching the underlying storage manager, we compared three external storage employed in our optimized backward chaining reasoner on I/O time. For all 14 queries from LUBM, the three storage managers SDB, TDB and OWLIM-SE, all have same number of accesses (calls to the Data-retrieval function) to the underlying store.

Based on this observation, we show in Table XIV the I/O time per access for SDB, TDB and OWLIM-SE using LUBM(50). The I/O time per store access of SDB is dramatically longer than both TDB and OWLIM-SE through all 14 queries in LUBM. From query 1 to 5 and query 13, the I/O time per store access of TDB is slightly longer than OWLIM-SE. For the other queries, TDB has shorter I/O time per store access. In general, TDB and OWLIM-SE have the similar performance in terms of I/O time.

D. Overall Performance

Finally, we consider the overall performance of our optimized backward chaining reasoner when based upon each of the three storage managers.

In order to compare the general performance of three triple store when employed in our optimized backward chaining reasoner, for all 14 queries from LUBM, we perform a comparison among SDB, TDB and OWLIM-SE on query response time using LUBM(50) in Table XV .

In Table XV, for LUBM (50), from query 1 to query 3 and query 6, OWLIM-SE has the fastest response time. Jena SDB + PostgreSQL performs fastest only for query 4, because the I/O time of Jena SDB is the longest out of three stores. For the rest of the queries, Jena TDB is fastest.

In Table XVI, we show a similar comparison of TDB and OWLIM-SE on query response time using LUBM(100). SDB was omitted from this comparison because the loading time of SDB is prohibitively long.

TABLE XIV .COMPARISON AMONG SDB, TDB AND OWLIM-SE AS EXTERNAL STORAGE ON I/O TIME PER STORE ACCESS

	LUBM(50)			
Number of facts (triples)	6,890,640			
	Time (ms), SDB+ Post-greSQL	***Time (ms), TDB***	***Time (ms), OWLIM-SE***	***#of Number of access to store***
Query1	41.42	2.32	0.70	132
Query2	50.76	0.48	0.35	353
Query3	1.63	0.42	0.28	65
Query4	82.38	0.38	0.14	455
Query5	1.57	0.36	0.20	81
Query6	298.74	0.67	5.12	153
Query7	237.69	0.13	0.52	286
Query8	72.24	0.07	0.43	917
Query9	221.45	0.02	0.17	351
Query10	223.33	0.07	0.14	218
Query11	2.08	0.05	0.12	616
Query12	2.07	0.03	0.10	2792
Query13	1.28	0.21	0.13	86
Query14	111.76	0.03	0.22	67

TABLE XV.COMPARISON BETWEEN SDB, TDB AND OWLIM-SE AS EXTERNAL STORAGE ON QUERY RESPONSE TIME

	LUBM(50)		
Number of facts (triples)	6,890,640		
	Clock Time		
	Time (ms), SDB+PostgreSQL	***Time (ms), TDB***	***Time (ms), OWLIM-SE***
Query1	6430	13440	3549
Query2	24960	36102	17046
Query3	406	58	61
Query4	46400	71298	45680
Query5	533	78	156
Query6	59144	32590	30470
Query7	83799	34580	45527
Query8	85563	48307	53013
Query9	95992	34583	49566
Query10	63100	20191	27916
Query11	3466	528	876
Query12	16253	2403	3199
Query13	374	39	37
Query14	8581	4731	5364

TABLE XVI.COMPARISON BETWEEN SDB, TDB AND OWLIM-SE AS EXTERNAL STORAGE ON QUERY RESPONSE TIME

	LUBM(100)	
Number of facts (triples)	13,405,677	
	Time (ms), TDB	***Time (ms), OWLIM-SE***
Query1	2652	5085
Query2	13884	29657
Query3	31	46
Query4	49109	82664
Query5	46	78
Query6	26020	51277
Query7	39873	76752
Query8	58609	98343
Query9	46925	85456
Query10	26894	52821
Query11	452	826
Query12	920	1716
Query13	15	31
Query14	7222	11263

In Table XVI, for LUBM(50), Jena TDB has better performance through all 14 queries. In general, our optimized backward chaining reasoner and external storage Jena TDB has the best performance especially when the size of the knowledge base increases.

VIII. Conclusion and Future Work

As knowledge bases proliferate on the Web, it becomes more plausible to add reasoning services to support more general queries than simple retrievals. In this paper, we have addressed a key issue of the large amount of information in a semantic web of data about science research. Scale in itself is not really the issue. Problems arise when we wish to reason about the large amount of data and when the information changes rapidly. In this paper, we report on our efforts to use backward-chaining reasoners to accommodate the changing knowledge base. We developed a query-optimization algorithm that will work with a reasoner interposed between the knowledge base and the query interpreter. We performed experiments, comparing our implementation with traditional backward-chaining reasoners and found, on the one hand, that our implementation could handle much larger knowledge bases and, on the other hand, could work with more complete rule sets (including all of the OWL rules). When both reasoners produced the same results our implementation was never worse and in most cases significantly faster (in some cases by orders of magnitude).

The analysis of reasoning over a large knowledge base that requires external storage has shown that no one component (backward chaining, I/O, middleware) dominates performance and thus improvements to any one of the three major components of the system will have only modest effect on the total time.

We have also addressed the issue of being able to scale the knowledge base to the level forward-chaining reasoners can handle. Preliminary results indicate that we can scale up to real world situations such as 6 Million triples. Optimizing the backward-chaining reasoner, together with the query-optimization allows us to actually outperform forward-chaining reasoners in scenarios where the knowledge base is subject to frequent change.

Although 6 million triples remains a modest size for a knowledge base, we believe that the key performance limitation is associated with the number of triples that are being brought into memory as intermediate results during the reasoning for a specific query. In [37] we tie the use of reasoning to a concept of "trust" reflecting changes made to the knowledge base since its last instantiation. Trust can be exploited to decide what goals arising during evaluation of a query require reasoning and what can be resolved by immediate lookup. The net effect is that considerably larger knowledge bases can be handled by limiting the scope of backward chaining to portions of the knowledge base untrusted due to recent changes.

Assessing the impact of using external storage on the individual optimization techniques produced in both of the cases we analyzed the same result. Having an external triple store magnified the effect of our optimization techniques. When we analyzed storage access we found that SDB was significantly slower than TDB and OWLIM-SE. The latter two had about the same performance. As the size of the knowledge base kept increasing the advantage of using Jena TDB with our optimized backward-chaining algorithm became more pronounced.

We will explore in future work ways to minimize in our backward chaining algorithms the number and size of requests for input from the underlying store and to employ caching techniques.

References

[1] H. Shi, K. Maly, and S. Zeil, "Query optimization in cooperation with an ontological reasoning service," The Fifth International Conferences on Advanced Service Computing (SERVICE COMPUTATION 2013), IARIA XPS Press, May-Jun. 2013, pp. 26-32.

[2] S.J.Russell and P. Norvig, Artificial intelligence: a modern approach., 1st ed.. Upper Saddle River: Prentice hall, pp. 265–275, 1995.

[3] H. Shi, K. Maly, and S. Zeil, "Optimized backward chaining reasoning system for a semantic web," Proc. The Fourth International Conference on Web Intelligence, Mining and Semantics (WIMS'14), ACM Press, June 2014.

[4] Microsoft. *Microsoft Academic Search*. [Online]. Available from: http://academic.research.microsoft.com/ 2014.02.25

[5] Z. Nie, Y. Zhang, J. Wen, and W. Ma, "Object-level ranking: bringing order to web objects," The 14th international World Wide Web conference (WWW2005), ACM Press, May 2005, pp. 567–574, doi:10.1145/1060745.1060828.

[6] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen., "Community information management," IEEE Data Engineering Bulletin, Special Issue on Probabilistic Databases, vol. 29, iss. 1, pp. 64–72, Mar. 2006.

[7] J.Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," Proc. Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2008), ACMPress, Aug. 2008, pp. 990–998, doi:10.1145/1401890.1402008.

[8] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, and C. Becker, "DBpedia-a crystallization point for the Web of Data," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, iss. 3, pp. 154–165, Sep. 2009, doi:10.1016/j.websem.2009.07.002.

[9] F. Suchanek, G. Kasneci, and G. Weikum, "Yago: a large ontology from wikipedia and wordnet," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 6, iss.3, pp.203–217, Sep. 2008, doi:10.1016/j.websem.2008.06.001.

[10] B. Aleman-Meza, F. Hakimpour, I. Arpinar, and A. Sheth, "SwetoDblp ontology of Computer Science publications," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, iss. 3, pp. 151–155, Sep. 2007, doi:10.1016/j.websem.2007.03.001.

[11] H. Glaser, I. Millard, and A. Jaffri, "Rkbexplorer. com: a knowledge driven infrastructure for linked data providers," The Semantic Web: Research and Applications, vol. 5021, pp. 797–801, Jun. 2008, doi:10.1007/978-3-540-68234-9_61.

[12] A. Kiryakov, D. Ognyanov, and D. Manov, "OWLIM–a pragmatic semantic repository for OWL," Proc. 6th international conference on Web Information Systems Engineering (WISE'05), Springer-Verlag, pp. 182-192, Nov. 2005, doi:10.1007/11581116_19.

[13] Oracle Corporation. 2013. *Oracle Database 11g R2*. [Online]. Available from: http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/ 2014.02.25

[14] O. Erling and I. Mikhailov, “RDF support in the Virtuoso DBMS,” Networked Knowledge-Networked Media, vol. 221, pp.7-24, 2009, doi:10.1007/978-3-642-02184-8_2.

[15] The Apache Software Foundation. *Apache Jena*. [Online]. Available from: http://jena.apache.org/ 2014.02.25

[16] Y.E. Ioannidis, “Query optimization,” ACM Computing Surveys (CSUR), vol. 28, iss. 1, pp. 121-123, March 1996, doi:10.1145/234313.234367.

[17] Semanticweb.org. *SPARQL endpoint*. [Online]. Available from:http://semanticweb.org/wiki/SPARQL_endpoint 2014.02.25

[18] W3C. *SparqlEndpoints*. [Online]. Available from: http://www.w3.org/wiki/SparqlEndpoints 2014.02.25

[19] M. Stocker, A. Seaborne, A. Bernstein, C. Kiefer, and D.Reynolds, “SPARQL basic graph pattern optimization using selectivity estimation,” The 17th international conference on World Wide Web (WWW 2008), ACM Press, pp. 595–604, Apr. 2008, doi:10.1145/1367497.1367578.

[20] O. Hartig and R. Heese, “The SPARQL query graph model for query optimization,” Proc. 4th European conference on the Semantic Web: Research and Applications (ESWC '07), Springer-Verlag, pp. 564-578, Jun. 2007, doi:10.1007/978-3-540-72667-8_40.

[21] W. Le, “Scalable multi-query optimization for SPARQL,” Proc. IEEE 28th International Conference on Data Engineering (ICDE 2012), IEEE Press, pp. 666–677, Apr. 2012, doi:10.1109/ICDE.2012.37.

[22] Y. Guo, Z. Pan, and J. Heflin, “LUBM: a benchmark for OWL knowledge base systems,” Web Semantics: Science, Services and Agents on the World Wide Web, vol. 3, iss. 2-3, pp.158–182, Oct. 2005, doi:10.1016/j.websem.2005.06.005.

[23] M. Kitsuregawa, H. Tanaka, and T. Moto-Oka, “Application of hash to data base machine and its architecture,” New Generation Computing, vol. 1, iss.1, pp. 63–74, Mar. 1983, doi:10.1007/BF03037022.

[24] H. Tamaki and T. Sato, “OLD resolution with tabulation,” Proc. Third international conference on logic programming, Springer, pp. 84-98, July 1986, doi:10.1007/3-540-16492-8_66.

[25] W3C. *OWL web ontology language reference*. [Online]. Available from: http://www.w3.org/TR/owl-ref/ 2014.02.25

[26] P. Hayes and B. McBride. *RDF semantics*. [Online]. Available from: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/ 2014.02.25

[27] H. Horst, “Combining RDF and part of OWL with rules: Semantics, decidability, complexity,” Proc. 4th International Semantic Web Conference (ISWC 2005), Springer, pp. 668-684, Nov. 2005, doi:10.1007/11574620_48.

[28] H. Shi, K. Maly, S. Zeil, and M. Zubair, “Comparison of ontology reasoning systems using custom rules,” International Conference on Web Intelligence, Mining and Semantics, ACM Press, May 2011, doi: 10.1145/1988688.1988708.

[29] K. Marriott and P. J. Stuckey, Programming with constraints: an introduction. Cambridge: MIT press, 1998.

[30] J. Santos and S. Muggleton, "When does it pay off to use sophisticated entailment engines in ILP?," in Inductive Logic Programming, P. Frasconi and F. A. Lisi, Eds. Heidelberg: Springer, pp. 214-221, 2011.

[31] Ontotext. *Owl-sameAs-optimization*. [Online]. Available from: http://www.ontotext.com/owlim/owl-sameas-optimisation 2014.02.25

[32] J. Lloyd, “Foundations of Logic Programming,” 2nd extend ed.. Springer-Verlag: Berlin, 1987.

[33] R. Kowalski and D. Kuehner, “Linear resolution with selection function,” Artificial Intelligence, vol. 2, iss. 3, pp. 227-260, 1972, doi: 10.1016/0004-3702(71)90012-9.

[34] Ontotext. *OWLIM-SE*. [Online]. Available from: http://owlim.ontotext.com/display/OWLIMv43/OWLIM-SE 2014.02.25

[35] The Apache Software Foundation. *SDB - persistent triple stores using relational databases*. [Online]. Available from: http://jena.apache.org/documentation/sdb/ 2014.02.25

[36] The Apache Software Foundation. *TDB*. [Online]. Available from: http://jena.apache.org/documentation/tdb/ 2014.02.25

[37] H. Shi, K. Maly, and S. Zeil, “Trust and hybrid reasoning for ontological knowledge bases,” Proc. the companion publication of the 23rd international conference on World wide web companion (WWW Companion '14), International World Wide Web Conferences Steering Committee, pp. 1189-1194, April 2014, doi: 10.1145/2567948.2579033.

Determining Robustness of Synchronous Programs under Stuttering

Ingram Bondin and Gordon J. Pace
University of Malta
Malta
ingrambondin@gmail.com, gordon.pace@um.edu.mt

***Abstract*—Robustness of embedded systems under potential changes in their environment is crucial for reliable behaviour. One typical environmental impact is that of the inputs being slowed down — due to which, the system may no longer satisfy its specification. In this paper, we present a framework for analysing the behaviour of synchronous programs written in Lustre under such environmental interference. Representing slow input by stuttering, we introduce both strong and weak slowdown robustness constraints with respect to this phenomenon. Furthermore, static and dynamic algorithmic techniques are used to deduce whether such constraints are satisfied, and the relationship between stateful programs and the slowdown model considered is explored.**

***Keywords*–Synchronous Languages; Lustre; Slowdown; Stutter**

I. Introduction

Software is increasingly becoming more prominent as a controller for a variety of devices and processes. Embedded systems operate within an environment, by which they are affected and with which they interact — this tight interaction usually means that changes to the environment directly change the behaviour of the embedded system. One such situation can occur when the environment slows down its provision of input to the system, possibly resulting from a variety of reasons. For example, the system producing the inputs or the communications channel on which these inputs pass to the program might be under heavy load, delaying the inputs; or the program is deployed on a faster platform, therefore making the input relatively slower.

One question that arises immediately in such scenarios is how the system behaves when its input slows down. Does it act in an expected manner, or does the slow input cause it to produce unwanted output? In this paper (which is an extended version of [1]), we develop an approach to study whether a system continues to behave correctly when the inputs are slowed down. This leads us to different notions of robustness to slow input since, for instance, in some cases we may desire the output to be delayed by the same amount as the inputs, whereas in others, the values but not the actual delays on the outputs are important.

The theory we develop is applied to the synchronous language Lustre [2], which enables the static deduction of a program's resource requirements, making it ideal for the design of embedded systems. Although retiming analysis techniques for continuous time can be found in the literature [3], our approach adapts them for discrete time, the timing model used by Lustre and other synchronous languages.

Such a theory requires addressing a number of considerations. In Section II we define streams [4], which are infinite sequences of values, as well as the Lustre programs that manipulate them. In the model we adopt, streams can be slowed down through the repetition of values, which is also called stuttering. Stuttering can be a valid model for slow input under several scenarios:

- If a memory's clock signal becomes slower, the memory will take more time to read new input, and thus will maintain its present output for a longer time. A program that samples the values of this memory at the same rate will then experience repetition in its input.
- The system providing the input might not be ready to provide its output, or it might experience a fault from which it needs time to recover. In these situations, some systems might keep their present output constant until they are ready once again. In this case, the receiving program will also experience repetition in its input.
- A physical process that is being sampled in order to provide input to a program might slow down. Under certain sampling conditions, the resulting input received by the program corresponds to experiencing stutter in its inputs.

The effect of slow input on program behaviour when modelled through stutter, provides a more complex scenario than that considered in the literature [3], [4], [5], [6], [7]. In particular, additional input symbols from a slow stream can change a program's internal state, requiring a more complex analysis to determine whether it behaves correctly. Our use of the Lustre language (with its simple semantics) will provide a useful setting for studying how program state and this type of slow input interact.

In Section III we identify a number of robustness properties that characterise acceptable program behaviour to slow input in a number of different scenarios. We consider both properties similar to those found in simpler slowdown models as well as weaker ones which are useful in our kind of model.

Given a robustness property, one desires an algorithmic way of checking whether or not it holds for a given program. Sections IV and V address this issue. Section IV considers a method based on the static analysis of the program's text, and yields compositionality results for the properties being considered. On the other hand, Section V focuses on a method based on the dynamic analysis of a program's state space.

The analysis techniques we present focus on Boolean (or finite type) Lustre programs. Although a number of results can be lifted to programs over arbitrary types, this does not apply in general and will be regarded as outside the scope of this work.

To demonstrate the application of the analysis techniques, we apply the two aforementioned approaches to various Boolean Lustre programs in Section VI. Finally, in Section VII we compare our work to existing results, while in Section VIII we give some concluding remarks. It should be noted that this paper is an extended version of [1]. The main extensions are the following.

- In Section III, we provide a proof that all robustness properties considered are relaxations of a strong property known as stretch robustness.
- In Section IV, we provide detailed proofs for all the static analysis results stated.
- In Section V, we give a new strong condition for showing that a program is not stretch robust, and a new condition for showing that a program is stretch robust without relying on the program being effectively stateless. This resolves the question of whether there are non-trivial stateful programs that satisfy stretch robustness when using a stutter based slowdown model.
- In Section VI, we apply the new conditions discussed in Section V to the programs considered in [1].

II. Streams, Slowdown and Lustre Programs

We adopt the standard view of a stream s as an infinite sequence of values over a particular type, representing the value of the stream over a discrete time domain. We shall write $s(t)$ to denote the value taken by stream s at time t.

Given a number of streams $s_1, \ldots, s_n$ we will find it convenient to collect these into a vector of streams $v = \langle s_1, \ldots, s_n \rangle$. In this case, we shall use the notation $v(s_i)$ to denote the stream s_i in the vector. Two vectors v_1 and v_2, can be combined, modulo renaming of streams, into one vector $v_1 \cup v_2$, containing all the streams from these two vectors.

Assuming a vector of streams $v = \langle s_1, \ldots, s_n \rangle$, we denote the behaviour of all streams at a particular time t by $v(t)$, which will yield the tuple of values $(s_1(t), \ldots, s_n(t))$.

By slowing down a stream, one obtains the same sequence of values, but possibly with some of the values repeated a number of times, representing stutter. A slowdown can be characterised using a latency function — a total function that returns the number of times each value in the stream will stutter for. Given a stream s that is slowed down according to a latency function λ, one obtains the slowed down stream s_λ:

$$s_\lambda = \underbrace{s(0), \ldots s(0)}_{\lambda(0)+1}, \underbrace{s(1), \ldots s(1)}_{\lambda(1)+1}, \ldots \underbrace{s(n), \ldots s(n)}_{\lambda(n)+1} \ldots$$

Note that s_λ is obtained from s by replacing the value of s at time t by a block of of $\lambda(t)+1$ copies of this value. We will write Start_t^λ to denote the time instant at which the t^{th} such block begins: $\sum_{i=0}^{t-1} \lambda(i)$. Similarly, End_t^λ denotes the time instant at which the block ends and is analogously defined.

Note that the constant zero latency function leaves the stream untouched. If a latency function is a constant function, we shall refer to it as *uniform*.

As before, we will extend this notation for vectors of streams, with $\langle s_1, \ldots s_n \rangle_\lambda$ being equivalent to $\langle s_{1\lambda}, \ldots, s_{n\lambda} \rangle$. From this it is easy to derive the useful fact that latency functions distribute over vector union, giving us the identity $(v_1 \cup v_2)_\lambda = (v_{1\lambda} \cup v_{2\lambda})$.

Lustre [2] provides a way of symbolically specifying systems that process streams in a declarative manner. A Lustre program $P = \langle V, I, O, E \rangle$ is defined over a set of stream variables V, with two disjoint subsets I and O consisting of the input and output stream variables of the program, respectively, and a set of equations E that explains how to compute the value of each output variable at every instant of time in terms of other program variables. Equations can take one of the following forms:

$$\begin{array}{rcl} y & = & \otimes(x_1, \ldots x_n) \\ y & = & \mathit{pre}\ x_1 \\ y & = & x_1 \texttt{ -> } x_2 \\ y & = & x_1\ \mathit{fby}\ x_2 \end{array}$$

Instantaneous operators $\otimes$ are used to represent computation performed at each time instant. For instance, the equation $y = \wedge(x_1, x_2)$ would update the value of stream variable y with the value of the conjunction of the stream variables x_1 and x_2 at each time instant: $y(t) = x_1(t) \wedge x_2(t)$. The *delay* operator *pre* allows access to the previous value of a given stream variable: $(\mathit{pre}\ x)(t+1) = x(t)$ with the resulting stream being undefined for the initial time point, at which it is said to take the value *Nil*. In fact, *pre* behaves like an uninitialised memory. The *initialisation* operator $x_1 \texttt{ -> } x_2$ yields a stream behaving like x_1 at the first time instant, and like x_2 elsewhere: $(x_1 \texttt{ -> } x_2)(0) = x_1(0)$ and $(x_1 \texttt{ -> } x_2)(t+1) = x_2(t+1)$. These last two operators are frequently combined to produce an initialised memory using the *followed-by* operator, with $x_1\ \mathit{fby}\ x_2$ being equivalent to $x_1 \texttt{ -> } \mathit{pre}\ x_2$.

Below we illustrate two sample programs. The program Toggle represents a toggle switch that starts in the Boolean state *true*, and which outputs its present state if its toggle input is *false* and inverts and outputs its present state if the toggle input is *true*. On the other hand, the program Siso is a 4-bit serial in serial out register, which starts with all its memories set to true.

```
node TOGGLE(toggle : bool)
returns(out : bool);
var X, Y : bool;
let
   out = if toggle then x else y;
   x = not y;
   y = true fby out
tel;
```

```
node SISO(i1 : bool)
returns(i5 : bool);
var i2, i3, i4 : bool;
let
   i2 = true fby i1;
   i3 = true fby i2;
   i4 = true fby i3;
   i5 = true fby i4;
tel;
```

We will use the notation P_{inst}, P_{delay}, P_{init}, and P_{fby} for the primitive programs with just one equation consisting of a single application of an instantaneous, delay, initialisation or followed-by operator, respectively. For each primitive program, the variable occurring on the left hand side of its equation is an output variable, those appearing on the right are inputs.

For a Lustre program P, $\text{dep}_0(P) \subseteq V \times V$ relates a stream variable y to a stream variable x if y is defined in P by an equation with x appearing on the right hand side. The irreflexive transitive closure of this relation denotes the dependencies between the stream variables and is written as $\text{dep}(P)$. Another important concept is that of an instantaneous dependency relation. This relation can be obtained by starting from the relation $\text{inst}_0(P) \subseteq V \times V$, which relates a stream variable y to a stream variable x only if y's defining equation involves x, and x does not appear in a pre equation or on the right hand side of an fby equation. The irreflexive transitive closure of this relation, $\text{inst}(P)$ denotes the instantaneous dependencies between stream variables. A Lustre program P is said to be well-formed if none of its variables instantaneously depend on themselves: $\forall s \cdot (s, s) \notin \text{inst}(P)$.

Given two Lustre programs P_1 and P_2 (with inputs I_1, I_2 and outputs O_1, O_2, respectively) their composition, written $P_1 \mid P_2$, is the Lustre program whose equation set is the union of the equation sets of the respective programs. Its inputs are the inputs of either program not appearing as outputs of the other ($I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$), and vice versa for its outputs ($O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$). In particular, certain specific types of composition shall be referred to as follows:

- *Disjoint composition*, if $O_2 \cap I_1 = O_1 \cap I_2 = \emptyset$.
- *Composition without feedback*, if $O_2 \cap I_1 = \emptyset$ or $O_1 \cap I_2 = \emptyset$.
- *Fully connected composition*, if $O_2 \cap I_1 = \emptyset$ and $O_1 = I_2$, or conversely $O_1 \cap I_2 = \emptyset$ and $O_2 = I_1$.

Another important operation is that of *adding a feedback loop* to a program P by connecting an output y to an input x written $P[y \to x]$, provided that y does not depend in any way on x, that is, $(y, x) \notin dep(P)$. Adding a feedback loop can also be defined in terms of composition of the original program with the Lustre program $P' = \langle \{x, y\}, \{y\}, \{x\}, \{x = y\} \rangle$, as follows:

$$P[y \to x] \stackrel{\text{df}}{=} P \mid P'$$

Assuming the existence of an ordering on the program's variables, given a Lustre program P, and a vector i that assigns a stream to each of the program's input variables, $P(i)$ denotes the vector o of output streams corresponding to the output variables of P as computed by the semantics of Lustre [2].

Our goal is therefore that of identifying Lustre programs P such that upon slowing down their inputs i according to a latency function λ, will result in P still being well behaved. In the next section we will identify different forms of robustness of $P(i_\lambda)$ with respect to the unslowed behaviour $P(i)$.

Boolean Lustre programs can also be compiled into automata spanning over the state space they cover [8]. This can be defined for Lustre programs using fby (instead of delays) as follows:

Definition 1: (Lustre Automaton). Let P be a Boolean Lustre program with n input variables, m output variables, and k fby equations of the form $y = x_1$ *fby* x_2. Then, this program can be compiled into an automaton $A = \langle S, s_{init}, \tau, \delta \rangle$, where S is its set of states, s_{init} is its initial state, $\tau : \mathbb{B}^n \times S \to S$ is its transition function and $\delta : \mathbb{B}^n \times S \to \mathbb{B}^m$ is its output function. The automaton, processes the input vector provided to the program one tuple at a time. During each instant, it uses its current input tuple and its present state to (i) move to a new state under the guidance of its transition function τ and (ii) output an output tuple as defined by its output function δ, which represents the values of the program's output variables at that particular time instant. The program P can be converted into automaton A using the following procedure.

External Initialisation: A program is said to be initialised externally if in at least one of its fby statements x_1 *fby* x_2, the initial variable x_1 depends on one of the program's input variables.

States: Each fby statement x_1 *fby* x_2 corresponds to a memory element in the program, whose value is determined by the variable x_1 at the first instant and by the variable x_2 at all further instants. Since each such memory can either be true or false, we create 2^k states, with each state representing one possible configuration of the program's memories. If the program is initialised externally, we also add a special initial state $init$ to the set of states.

Initial State: If the program is initialised externally, the initial state is $init$. Otherwise, the initial state is the state corresponding to the configuration obtained by evaluating the variables of the form x_1 within the program's fby statements.

Transition Function: With n input variables, there are 2^n possible input tuples. Each state therefore has 2^n transitions, with each transition labelled with the associated input tuple. Given a state $s \neq init$ and input tuple a, the next state $\tau(a, s)$ is computed as follows (i) assign the configuration represented by present state s to the respective variables of the form x_2 occurring on the right hand side of *fby* statements, (ii) assign the input values represented by tuple a to the respective input variables and (iii) simulate the Lustre program, using the defining equations of the variables of the form x_2 to determine the configuration of the memories at the next time instant, allowing the selection of the appropriate next state. The initial state $init$, if present, also has 2^n transitions. The next states are determined as follows (i) assign the input values represented by tuple a to the respective input variables, (ii) use the defining equations of variables of the form x_1 to compute the value of the initialisation variables and (iii) simulate the Lustre program using the defining equations of the variables of the form x_2, which determine the next state. Again, these values determine the configuration of the memories at the next time instant and allow the selection of the appropriate next state.

Output Function: Each transition is associated with an m-tuple, which represents the values of the output variables when the automaton finds itself in a certain state and processes a certain input tuple. The procedure for obtaining the output tuple is similar to that for obtaining the next state, except that the output tuple is constructed by simulating the program and considering the values of the output variables.

Figure 1 shows the automaton that would be obtained by applying the above procedure to the toggle switch program TOGGLE. The two states represent the two possible configurations that the memory corresponding to the program's only *fby* equation can be in. Meanwhile, for each transition, the value on the left shows the value of the toggle input variable that causes the transition, and the value on the right shows the output value computed by the program. We shall return to this representation of the TOGGLE program at a later stage.

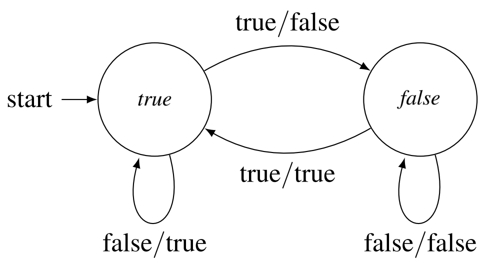


Figure 1. Automaton obtained from toggle switch program

We now consider a number of different forms of program robustness to slow input.

III. SLOWDOWN ROBUSTNESS

Whether a program behaves in an acceptable way depends on the scenario it is operating in. In this section, the four robustness properties of *stretch robustness*, *stutter robustness*, *fast-enough robustness* and *immediate-at-first robustness* are introduced, characterising desirable behaviour under different circumstances.

A. Stretch Robustness

Stretch robustness (STR) specifies the fact that if the input of a program slows down by some amount, then the output of a program should slow down by the same amount. This property can be formalised by requiring that whenever a latency function λ is applied to a program's input, the program will respond by applying the same latency to its output.

Definition 2: (Stretch Robustness). A program P is said to be *stretch robust with respect to a latency function* λ, if for any input vector i: $P(i_\lambda) = P(i)_\lambda$. P is simply said to be *stretch robust* if it is stretch robust with respect to all latency functions.

The figure below shows the relationship between a slow input vector i_λ and the required program output $P(i_\lambda)$:

$$\begin{array}{c|ccccc}
i_\lambda & \underbrace{i(0),\dots i(0)}_{\lambda(0)+1}, & \underbrace{i(1),\dots i(1)}_{\lambda(1)+1}, & \dots & \underbrace{i(n),\dots i(n)}_{\lambda(n)+1} & \dots \\
\hline
P(i_\lambda) & \underbrace{o(0),\dots o(0)}_{\lambda(0)+1}, & \underbrace{o(1),\dots o(1)}_{\lambda(1)+1}, & \dots & \underbrace{o(n),\dots o(n)}_{\lambda(n)+1} & \dots
\end{array}$$

One immediate consequence of this property is that additional repetition of the program's input does not cause the program to change its output. Stretch robustness is thus useful in situations where one requires the program not to change its output when faced with additional latency. Stretch robustness is a very strong property, which can be relaxed in a number of ways to obtain weaker criteria that may be sufficient in certain circumstances. We shall now consider these criteria, leaving the proof that these are indeed relaxations of stretch robustness to Theorem 6.

B. Stutter Robustness

Stutter robustness (STU) requires that if the input of a program slows down by some amount, the output of the program should also slow down, but possibly at a different rate. This will be modelled by requiring that whenever a latency function λ is applied to a program's input, the program will respond by applying *some* latency function λ' to its output. Unlike stretch robustness, λ and λ' need not be equal:

Definition 3: (Stutter Robustness). A program P is *stutter robust with respect to a latency function* λ if there exists a latency function λ' such that for every input vector i: $P(i_\lambda) = P(i)_{\lambda'}$. P is said to be *stutter robust* if it is stutter robust with respect to any latency function.

The relationship between a slow vector of inputs i_λ and the required program output $P(i_\lambda)$ is shown below:

$$\begin{array}{c|ccccc}
i_\lambda & \underbrace{i(0),\dots i(0)}_{\lambda(0)+1}, & \underbrace{i(1),\dots i(1)}_{\lambda(1)+1}, & \dots & \underbrace{i(n),\dots i(n)}_{\lambda(n)+1} & \dots \\
\hline
P(i_\lambda) & \underbrace{o(0),\dots o(0)}_{\lambda'(0)+1}, & \underbrace{o(1),\dots o(1)}_{\lambda'(1)+1}, & \dots & \underbrace{o(n),\dots o(n)}_{\lambda'(n)+1} & \dots
\end{array}$$

Thus for a stutter robust program, the output under slow input can be obtained from the original output by adding *any* number of repetitions to the values appearing in the original output, without adding any other artifacts nor removing any values. This means that stutter robustness is useful in situations where one needs to ensure that the output under slow input has the same structure as the original output, but one is able to tolerate additional repetition in the slow output.

C. Fast-Enough and Immediate-at-First Robustness

In stretch robustness, the value of the outputs remains equal to the original value in the unslowed system. In fast-enough robustness (FE) this constraint is relaxed by requiring only that the program converge to the original output before the slowed down input ends. Formally, we shall say that a program is fast-enough robust if, when we apply a latency function λ to the program's input, the slow output has the property that its value at the end of each block of repetitions (at points of the form End_t^λ) is equal to the value taken by the original output at the points t (i.e., those points that were expanded into the blocks of repetitions).

Definition 4: (Fast-Enough Robustness). A program P is *fast-enough robust with respect to a latency function* λ if for any input vector i:

$$\forall t : \mathbb{T} \cdot P(i_\lambda)(End_t^\lambda) = P(i)(t)$$

P is said to be *fast-enough robust* if it is fast-enough robust with respect to any latency function.

Fast-enough robustness will be primarily of interest for particular latency functions. As a property, it can be visualised as follows (using ? to indicate don't-care values):

$$\begin{array}{c|ccccc}
i_\lambda & \underbrace{i(0),\ldots i(0)}_{\lambda(0)+1}, & \underbrace{i(1),\ldots i(1)}_{\lambda(1)+1}, & \ldots & \underbrace{i(n),\ldots i(n)}_{\lambda(n)+1} & \ldots \\
\hline
P(i_\lambda) & \underbrace{?,\ldots ?, o(0)}_{\lambda(0)+1}, & \underbrace{?,\ldots ?, o(1)}_{\lambda(1)+1}, & \ldots & \underbrace{?,\ldots ?, o(n)}_{\lambda(n)+1} & \ldots
\end{array}$$

This robustness property is useful in scenarios in which one can tolerate the fact that additional latency on the input might produce undesirable intermediate results as long as the original value is produced by the end of the latency period.

The dual of fast-enough robustness is *immediate-at-first robustness* (IAF) — instead of constraining the slow input to converge to the original value before a block of repetitions ends, it requires it to produce the original value as soon as a block of repetitions starts, leaving it free to assume any value until that block of repetition ends.

Definition 5: (Immediate-At-First Robustness). A program P is said to be *immediate-at-first robust with respect to latency function* λ if for any input vector i:

$$\forall t : \mathbb{T} \cdot P(i_\lambda)(Start_t^\lambda) = P(i)(t)$$

P is said to be *immediate-at-first robust* if it satisfies the above constraint with respect to any latency function.

Immediate-at-first robustness can be visualised as follows:

$$\begin{array}{c|ccccc}
i_\lambda & \underbrace{i(0),\ldots i(0)}_{\lambda(0)+1}, & \underbrace{i(1),\ldots i(1)}_{\lambda(1)+1}, & \ldots & \underbrace{i(n),\ldots i(n)}_{\lambda(n)+1} & \ldots \\
\hline
P(i_\lambda) & \underbrace{o(0), ?,\ldots ?}_{\lambda(0)+1}, & \underbrace{o(1), ?,\ldots ?}_{\lambda(1)+1}, & \ldots & \underbrace{o(n), ?,\ldots ?}_{\lambda(n)+1} & \ldots
\end{array}$$

This robustness property is useful in scenarios in which one requires the program to react immediately as soon as the latency on a previous input value wears off, but in which further repetition of the input can be safely ignored by outputting any result.

We now prove that the stutter robustness, immediate-at-first robustness and fast-enough robustness properties are relaxations of the stretch robustness property. To achieve this, it will be convenient to treat each property as the set of all Lustre programs that satisfy it.

Theorem 6: (Stutter robustness, immediate-at-first robustness and fast-enough robustness are relaxations of stretch robustness) $STR \subseteq STU, IAF, FE$.

Proof: We first show that $STR \subseteq STU$. Let P be a stretch robust program. We shall prove that it is also stutter robust. For program to be stutter robust, it must respond to a latency function λ on its input by applying some latency function λ' to its output. Since P is stretch robust, when a latency function λ is applied to its input, it will respond by applying an identical λ to its output. Hence P also satisfies stutter robustness.

To prove the other two inclusions, we first note that when P is stretch robust, its output under slow input, $P(i_\lambda)$ is organised into successive blocks, with the t^{th} block lying between $Start_t^\lambda$ and End_t^λ. In addition, the symbols within the t^{th} block are all of the form $P(i)(t)$, that is, equivalent to the t^{th} output of P under the unslowed input i. From this, it follows directly that $P(i_\lambda)$ at the points of the form $Start_t^\lambda$ is equal to $P(i)(t)$ as required by immediate-at-first robustness. It also follows that $P(i_\lambda)$ at the points of the form End_t^λ is equal to $P(i)(t)$, thus satisfying fast-enough robustness. Hence the directions $STR \subseteq IAF$ and $STR \subseteq FE$ are also proven. □

We now proceed to consider algorithmic means for detecting whether a Lustre program satisfies a robustness property.

IV. Detecting Robustness: Static Analysis

The first approach to checking whether a Boolean Lustre program satisfies a robustness property is based on a static analysis of the structure of the Lustre program. The analysis is based on two kinds of result: (i) Theorem 7, which identifies those primitive programs that satisfy certain robustness properties and; (ii) Theorems 8, 9, 10 and 11,which identify those robustness properties that are preserved upon composition of two robust programs. We now proceed to state these theorems.

Theorem 7: Primitive Lustre programs all come with a level of guaranteed robustness: (i) instantaneous programs are robust under all four forms; (ii) delay and followed-by programs are robust under stutter and immediate-at-first robustness; and (iii) primitive initialisation programs are immediate-at-first robust.

Proof: (i) Instantaneous programs apply a pointwise operator to their input streams to obtain their output streams. Thus, the same input tuple always causes the same output tuple. Repetition of inputs through latency will therefore cause repetition of outputs, which makes the program stretch robust. (ii) The output of delay and fby programs has an additional initial value with respect to the input stream. Slowing the input stream down by a latency function, causes the program to attach this value to the slow stream. The output under slow input can therefore be obtained from the original input through a latency function, which does not repeat the attached element, and which repeats all subsequent elements accordingly. These programs are therefore stutter robust. The programs are also immediate-at-first robust as can be inferred from the figure below, which shows how the values of the original output (first row) are associated to the corresponding blocks of the output under latency (second row). It is clear that the value at the beginning of each block is equal to the corresponding value in the original output.

$$\begin{array}{c|ccc}
P(i) & Nil & x_1(0) & x_1(1) \\
\hline
P(i_\lambda) & Nil, \underbrace{x_1(0)\ldots x_1(0)}_{\lambda(0)} & x_1(0), \underbrace{x_1(1)\ldots x_1(1)}_{\lambda(1)} & x_1(1), \underbrace{x_1(2)\ldots x_1(2)}_{\lambda(2)}
\end{array}$$

(iii) Initialisation programs take the first value of stream x_1, and attach to it the stream x_2 from its second value onwards. The figure below shows how the blocks of output under slow input, relate to the original output, illustrating the fact that the value at the beginning of each block is equal to the corresponding value in the original output.

$P(i)$	$x_1(0)$	$x_2(1)$	$x_2(2)$
$P(i_\lambda)$	$x_1(0), \underbrace{x_2(0)...x_2(0)}_{\lambda(0)}$	$\underbrace{x_2(1)...x_2(1)}_{\lambda(1)+1}$	$\underbrace{x_2(2)...x_2(2)}_{\lambda(2)+1}$

□

We shall now consider the effect of composing robust programs.

Theorem 8: The composition without feedback of stretch robust programs is stretch robust.

Proof: We consider two stretch robust programs P_1 and P_2 that are composed without feedback, as shown in Figure 2. As a remark we note that P_2 obtains some input from P_1 (the vector j) and some input from an external source (the vector i_2), which can be collectively represented by the vector $j \cup i_2$. On the other hand, the output of the complete system, consists of some output from P_1 (the vector o_1) and some output from P_2 (the vector o_2), which is collectively represented by the vector $o_1 \cup o_2$.

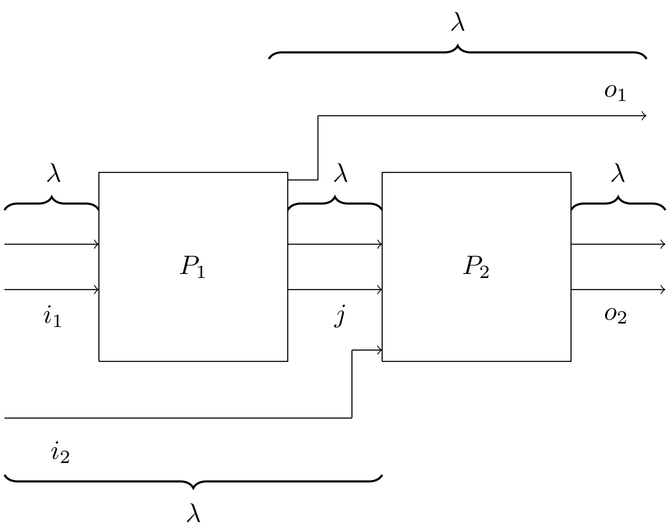


Figure 2. Composition without feedback preserves STR

Since the composed program reacts to input $i_1 \cup i_2$ with output $o_1 \cup o_2$, we aim to show that given slow input $(i_1 \cup i_2)_\lambda$, the program will react with the output $(o_1 \cup o_2)_\lambda$.

To see why this is true, we observe that P_1 is stretch robust, and thus will react to the input $i_{1\lambda}$ with the output vector $(o_1 \cup j)_\lambda$. Since latency functions distribute over the union of vectors, we can conclude that P_1 outputs two sets of streams, corresponding to the two output vectors $o_{1\lambda}$ and j_λ. P_2 will thus receive as input the vectors j_λ and $i_{2\lambda}$, which we can represent by a single input vector $(j_\lambda \cup i_{2\lambda})$ or equivalently, $(j \cup i_2)_\lambda$. Being stretch robust itself, it must then output the slow output vector $o_{2\lambda}$. Combining the output vectors of the system $o_{1\lambda}$ and $o_{2\lambda}$ we obtain the vector $(o_{1_\lambda} \cup \ o_{2_\lambda})$, which is equivalent to $(o_1 \cup \ o_2)_\lambda$ as expected. □

Theorem 9: Adding a feedback loop to a stretch robust program gives a stretch robust program.

Proof: Let P be a stretch robust program, and consider adding a feedback loop between one of its outputs y and one of its inputs x. Our definition of adding a feedback loop also requires y not to depend in any way on x, that is, $(y, x) \notin dep(P)$. This situation is shown in Figure 3, where the triangle shows that y can only be a function of a set of inputs $I^* \subseteq I$ that excludes x.

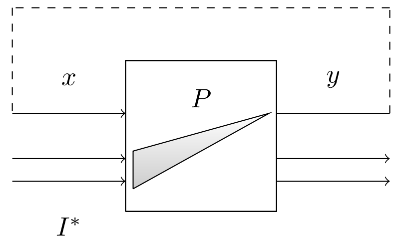


Figure 3. Adding a feedback loop to a program

To prove that this program remains stretch robust, we proceed by constructing an equivalent program that 'unwinds' the feedback loop, and which we can prove to be stretch robust by the application of Theorem 8. The new program will contain two copies of P, which we call P_1 and P_2. These are connected as shown in Figure 4.

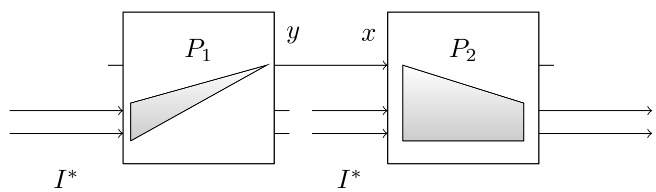


Figure 4. Unwinding the feedback loop

We shall use P_1 to compute the value of y from I^*, discarding all outputs but y. Following this, P_2 can be used to recompute all of the outputs, by using the inputs I^*, and by feeding the output y from P_1 to the input x in P_2 in order to simulate the feedback equation $x = y$. Since the feedback loop plays no further role in computing output of the original program, we simply discard y from the output of P_2 and keep the rest of the outputs. Note that in Figure 4, the parallelogram inside P_2 is meant to indicate that the outputs of P_2 that are relevant to the final computation can now depend on all of its inputs).

We now notice that i) P_1 and P_2 are copies of P, ii) P is stretch robust and iii) the construction reduces to a composition without feedback of stretch robust programs. By Theorem 8, the composition without feedback of stretch robust programs is stretch robust. Hence, the equivalent program with the feedback loop is also stretch robust, as required. □

Theorem 10: The fully connected composition of two stutter robust programs is stutter robust.

Proof: We consider two arbitrary stutter robust programs P_1 and P_2, and show that their fully connected composition is also stutter robust. Since P_1 and P_2 are being composed in a fully connected way, every output of P_1 is connected to an input of P_2, and there are no feedback connections. This is shown in Figure 5.

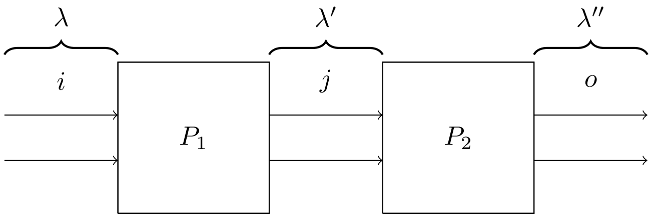


Figure 5. Fully connected composition preserves STU

Now suppose that if we pass a vector i to P_1, the program responds by outputting vector j. Also suppose that when P_2 receives vector j it outputs vector o in response. We need to show that if a latency function λ is applied to the composite program's input vector i, the composite program applies some latency function to its output vector o. Since P_1 is stutter robust, applying λ to the input vector i will make P_1 apply some latency function λ' to its output j. Hence, P_2 receives the vector j slowed down by λ' as input. Since P_2 is also stutter robust it will apply some other latency function λ'' to its output o. Thus, applying a latency function to the input of the composed program, causes the composed program to slow its output by some latency function, proving that stutter robustness is preserved by fully connected composition. □

Theorem 11: The disjoint composition of two immediate-at-first robust programs is immediate-at-first-robust.

Proof: We consider two immediate-at-first robust programs P_1 and P_2 in disjoint composition, as shown in Figure 6. We note that P_1 computes output vector $P_1(i_1)$ given input vector i_1, while P_2 computes output vector $P_2(i_2)$ when given input vector i_2. Thus, if the composed program is given the input vector $(i_1 \cup i_2)$, it will react with an output vector $(P_1(i_1) \cup P_2(i_2))$.

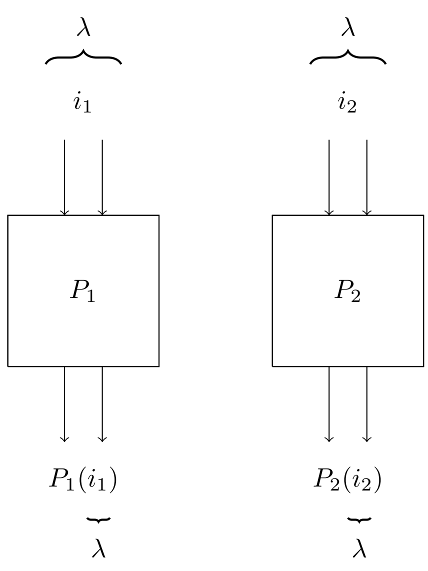


Figure 6. Disjoint composition preserves IAF

We now note that both P_1 and P_2 are immediate-at-first robust, and that under slow input, will generate two disjoint output vectors with the following properties. The output vector $P_1(i_{1\lambda})$ of P_1 will have the property that for every stream s:

$$s(Start_t^\lambda) = P_1(i_1)(s)(t) \tag{1}$$

while the output vector $P_2(i_{2\lambda})$ of P_2 will have the following property for every one of its streams s.

$$s(Start_t^\lambda) = P_2(i_2)(s)(t) \tag{2}$$

On the other hand, to prove that the composite program is immediate-at-first-robust, we would need to show that each of the streams s in its output vector $(P_1(i_{1\lambda}) \cup P_2(i_{2\lambda}))$ obeys the following property:

$$s(Start_t^\lambda) = (P_1(i_1) \cup P_2(i_2))(s)(t)$$

However, we note that we are always able to determine whether a stream s in $(P_1(i_{1\lambda}) \cup P_2(i_{2\lambda}))$ originates from P_1 or P_2. Hence, it is sufficient to ensure that such a stream s always satisfies the property

$$s(Start_t^\lambda) = P_n(i_n)(s)(t) \tag{3}$$

where n is the index of the program generating s. By equation (1), we know that property (3) holds when s is generated by P_1 (when n=1), while equation (2) ensures that property (3) holds when s is generated by P_2 (when n=2). The theorem is thus proved. □

We now consider a method that analyses the behaviour of the particular program under examination, rather than its structure.

V. Detecting Robustness: Dynamic Analysis

The theorems in Section IV allow us to conclude robustness of composed programs in a syntactically compositional manner. In this section, we give richer, although more expensive, semantic analysis techniques for Boolean Lustre programs allowing for dynamic robustness analysis of their behaviour. Through the use of symbolic methods, such as with Binary Decision Diagrams (BDDs), the analysis can be applied either on whole programs or to subprograms. In the latter case, the static analysis theorems can then be used to obtain results about the composition of the subprograms.

The techniques we shall discuss rely on identifying conditions on the Lustre automaton that are sufficient to guarantee that certain robustness properties are satisfied by that program. Two types of conditions are defined: (i) latency independent conditions, which check whether a robustness property holds in general, and (ii) latency dependent conditions, which check whether a property holds when some particular latency function is applied to the program's input.

The conditions identified can be checked using either an exhaustive analysis of the automaton's state space, or preferably using a symbolic representation of the automaton such as BDDs to ensure that the approach scales up to larger systems.

A. Latency Independent Conditions

We start by identifying properties that guarantee slowdown robustness for any latency function. The first condition we shall consider checks whether being in different states can cause the program's output to change. If this is not the case, the program is stretch robust.

Theorem 12: (Condition 1 — Stretch Robustness). Programs satisfying the following condition are stretch robust:

$$\forall a, s, s' \cdot \delta(a, s) = \delta(a, s')$$

Proof: Under such a condition, a particular input tuple always generates the same output tuple, independently of the state the automaton find itself in. Thus, any repetition of an input tuple caused by a latency function causes a repetition of the corresponding output tuple. This is sufficient to ensure stretch robustness. □

We now consider a condition that stops a program from being stretch robust. Condition 2 states that if we can find a state s, which under some tuple a moves to s', such that s and s' yield different outputs under a, then the program must fail to be stretch robust.

Theorem 13: (Condition 2 — Failure of Stretch Robustness) Programs satisfying the following condition are not stretch robust:

$$\begin{aligned} &\exists\, a, b, b', s, s' \cdot \\ &\quad \tau(a, s) = s' \wedge \delta(a, s) = b \\ &\quad \delta(a, s') = b' \wedge b \neq b' \end{aligned}$$

Proof: We show that the existence of reachable states s and s' described by the theorem is sufficient to find an input vector and a latency function λ such that the program fails to be stretch robust.

We first build the input vector as follows. Starting from the initial state of the automaton, we follow any path of some length n leading to state s. This path gives us the first n tuples of the input vector i_1. At state s, we then follow the transition for tuple a. The rest of the input vector can then be selected arbitrarily, yielding $i_1\ a\ i_2$.

Given the input vector $i_1\ a\ i_2$, the automaton will first process the tuples in i_1 and output the first n tuples of the output vector o_1. Since the automaton is now in state s, it will output tuple b, followed by some sequence of tuples o_2.

To prove the theorem, we select a latency function that adds one repetition at time instant $n+1$, and 0 repetitions elsewhere. Applying this latency function to the input vector thus gives $i_1\ a\ a\ i_2$.

On this slow vector the automaton will first output $o_1\ b$, as before, but being now in state s', will output some tuple $b' \neq b$, followed by the rest of the output vector o_3. It is clear that $o_1\ b\ b'\ o_3$ is different from $(o_1\ b\ o_2)_\lambda = (o_1\ b\ b\ o_2)$. Thus the program is not stretch robust, as required. □

Condition 1 indicates that programs whose states do not affect program output are stretch robust. The states in such programs are effectively redundant, in the sense that automata corresponding to these programs can be minimised (say, using the *partitioning minimisation procedure* [9]) into equivalent automata having just one state. Through condition 2, we also know that programs having two successive states yielding different output tuples under the same input tuple cannot be stretch robust. This raises the question of whether there are any non-trivial stateful programs that satisfy stretch robustness.

This question can in fact be given a positive answer by identifying a condition that lies between these two extremes. The intuition behind condition 3 is as follows. Given any state s, which under input a yields output b, we will allow it to transition to state s' if i) s' under a also yields b, thus escaping condition 2, and ii) s' self-loops under a. The effect of the second requirement is that s' becomes a 'sink' for repetitions of a, and preserves the output b experienced under s, guaranteeing stretch robustness. Note that as long as the requirements for transitions between states are satisfied, there can very well be states with different output behaviours. This is something that is not covered by condition 1.

Theorem 14: (Condition 3 — Stretch Robustness) Programs satisfying the following condition are stretch robust:

$$\begin{aligned} &\forall a, b, s, s' \cdot \\ &\quad (\tau(a, s) = s' \wedge \delta(a, s) = b) \\ &\implies (\tau(a, s') = s' \wedge \delta(a, s') = b) \end{aligned}$$

Proof: When processing its input vector i, the automaton will use the current input $i(t)$ and state $s(t)$, to compute the output $o(t)$ and next state $s(t+1)$. When input i has latency λ, the program receives consecutive blocks of constant inputs, with the n^{th} block consisting of tuples of the form $i(n)$. To satisfy stretch robustness, the program must apply λ to its output, thus yielding blocks of constant outputs such that the n^{th} block consisting of tuples of the form $o(n)$.

We observe that if the automaton finds itself in state $s(n)$ at the beginning of input block n, the condition guarantees that (i) at the first time instant in the block the automaton outputs $o(n)$ and moves to state $s(n+1)$; (ii) during the rest of the time instants in the block it will again output $o(n)$ and self loop to state $s(n+1)$ and (iii) it will find itself in state $s(n+1)$ at the start of the $(n+1)^{th}$ input block.

Noting that in the case of block 0, the automaton starts in the initial state $s(0)$, provides the base case for an inductive argument which guarantees that the automaton finds itself in state $s(n)$ at the start of the n^{th} block, causing the output to be $o(n)$ throughout that block as required. □

Although condition 3 shows that there are non-trivial stateful stretch robust programs, it also sets a strong requirement for this to be the case. We now investigate conditions for weaker robustness properties, starting from stutter robustness.

Under stutter robustness, slowing a program's input by a latency function λ, causes the program to slow its output by a latency function λ'. In practice, this means that the output under slow input can be obtained through the repetition of the original output tuples only. We show that if the automaton has a certain feature, then this property cannot hold.

Theorem 15: (Condition 4 — Failure Of Stutter Robustness). Programs satisfying the following condition are not stutter robust:

$$\begin{aligned}&\exists a, b, s, s', j, k, l \cdot \\ &\quad \delta(a, s) = j \wedge \tau(a, s) = s' \wedge \\ &\quad \delta(a, s') = k \wedge k \neq j \wedge \\ &\quad \delta(b, s') = l \wedge b \neq a \wedge l \neq k\end{aligned}$$

Proof: Condition 4 looks for the presence of reachable states s and s' having the following properties: (i) under input tuple a, state s outputs tuple j and passes to state s'; (ii) under input tuple a, state s' outputs $k \neq j$ and (iii) under input tuple $b \neq a$, state s' outputs tuple $l \neq k$.

We now show that if this structure is present in the automaton, there will always be some input vector and some latency function that breaks the stutter robustness property. We first construct the input vector as follows. Choose a path from the start state s_{init} to the state s. By following this path of n transitions, we obtain the first n tuples of the input vector, which we denote by i_1. We also obtain the first n tuples of the output vector, denoted by o_1. To this initial segment of the input vector, one appends the input tuples a b, which causes the resulting output vector to be augmented by the output tuples j l. The rest of the input vector i_2 can be chosen arbitrarily.

We now choose a latency function, which when applied to the input vector i_1 a b i_2 above, breaks the property. The chosen latency function will insert 1 repetition for the input tuple at time instant $n + 1$, and 0 repetitions elsewhere. Applying this latency function to the input vector chosen earlier will yield i_1 a a b i_2. Through the presence of the regularity identified in the theorem, the resulting output will be the initial segment of the output vector o_1 followed by the output tuples j k, which means that with respect to the original output an l tuple has been deleted. This fact alone makes it impossible to derive the output under slow input from the original output through the addition of repetitions only. □

Finally, we can also identify a sufficient condition for immediate-at-first robustness, which we can obtain by relaxing condition 3. We shall still insist that if a state s transitions to s' under some input, then it must loop in s' under that input. However, we will not require input in s and s' to yield the same output. As the proof shows this is enough to guarantee that the program is immediate-at-first robust.

Theorem 16: (Condition 5 — Immediate-At-First Robustness). Programs satisfying the following condition are immediate-at-first-robust:

$$\forall a, s, s' \cdot (\tau(a, s) = s') \implies (\tau(a, s') = s')$$

Proof: When processing an input vector i, the automaton uses the current input $i(t)$ and state $s(t)$, to compute the output $o(t)$ and next state $s(t+1)$. When input i has latency λ, the program receives consecutive blocks of constant inputs, with the n^{th} block consisting of tuples of the form $i(n)$. For the program to be immediate-at-first robust, the output at the beginning of the n^{th} block must have the form $o(n)$.

We observe that if the automaton finds itself in state $s(n)$ at the beginning of block n, the condition guarantees that (i) at the first time instant in the block the automaton moves to state $s(n+1)$; (ii) it stays in state $s(n+1)$ for the remainder of the block and (iii) the $(n+1)^{th}$ block starts in state $s(n+1)$. Noting that in block 0, the automaton starts in the initial state $s(0)$, provides the base case for an inductive argument which guarantees that the automaton finds itself in state $s(n)$ at the beginning of the n^{th} block, causing the output to be $o(n)$ as required. □

B. Latency Dependent Conditions

So far, we tried to identify programs that are robust under an input slowed down by an unknown latency. If one knows that the inputs of a program are going to slow down by some uniform latency function $\lambda(t) = c$, where c is a constant, it is possible to check whether the program is robust for that particular scenario using the following weakened conditions.

Condition 6 requires that for any state s, the state reached by the automaton after the occurrence of a specific input tuple, $\tau(a, s)$, is the same state reached after the occurrence of $c+1$ such input tuples, which we denote by $\tau^{c+1}(a, s)$.

Theorem 17: (Condition 6 — Immediate-At-First-Robustness). Programs satisfying the following condition for some positive natural number $c \geq 2$, are immediate-at-first robust for latency functions of the form $\lambda(t) = c$:

$$\forall a, s \cdot \tau(a, s) = \tau^{c+1}(a, s)$$

Proof: When processing an input vector i, the automaton uses the current input $i(t)$ and state $s(t)$, to compute the output $o(t)$ and next state $s(t + 1)$. When input i has latency λ, the program receives consecutive blocks of constant inputs of size $c+1$, with the n^{th} block consisting of tuples of the form $i(n)$. For the program to be immediate-at-first robust, the output at the beginning of the n^{th} block must have the form $o(n)$.

Suppose that at the beginning of the n^{th} block the automaton finds itself in state $s(n)$. Then at the beginning of the $(n + 1)^{th}$ block it is in state $s(n + 1)$ on account of (i) at the first time instant in the n^{th} block the automaton moves to $s(n + 1)$ and (ii) the condition guarantees that after $c + 1$ steps of the same input the automaton will return to $s(n+1)$. Noting that in block 0, the automaton starts in the initial state $s(0)$, provides the base case for an inductive argument which guarantees that the automaton finds itself in state $s(n)$ at the beginning of the n^{th} block, causing the output to be $o(n)$ as required. □

The final condition that will be considered requires that if an automaton is in state s, it will return to the same state s after c repetitions of the input.

Theorem 18: (Condition 7 — Immediate-At-First and Fast-Enough Robustness). Programs satisfying the following condition for some positive natural number $c \geq 2$, is immediate-at-first robust and fast-enough robust for latency functions of the form $\lambda(t) = c$:

$$\forall a, s \cdot \tau^{c}(a, s) = s$$

Proof: When processing an input vector i, the automaton uses the current input $i(t)$ and state $s(t)$, to compute the output $o(t)$ and next state $s(t+1)$. When input i has latency λ, the program receives consecutive blocks of constant inputs of size $c + 1$, with the n^{th} block consisting of tuples of the form $i(n)$. For the program to be immediate-at-first robust, the output at the beginning of the n^{th} block must have the form $o(n)$. Similarly, for a program to be fast-enough robust, the output at the end of the n^{th} block must have the form $o(n)$.

Suppose that at the beginning of the n^{th} block the automaton finds itself in state $s(n)$. Then at the end of the n^{th} block it is in state $s(n)$ on account of the fact that the automaton returns to its original state after c transitions of the same input. This state also combines with input $i(n)$ to ensure passage to state $s(n{+}1)$ at beginning of the $(n{+}1)^{th}$ block. Noting that in block 0, the automaton starts in the initial state $s(0)$, provides the base case for an inductive argument which guarantees that the automaton always finds itself in state $s(n)$ at the end of the n^{th} block, causing the output to be $o(n)$ as required for fast-enough robustness, and in state $s(n+1)$ at the beginning of the $(n+1)^{th}$ block guaranteeing that the output is $o(n+1)$ as required by immediate-at-first robustness. □

We now apply the static and dynamic analysis techniques discussed earlier via a case study.

VI. Case Study

The static and dynamic analysis theorems were applied to six Boolean Lustre programs to examine whether these are strong enough to deduce slowdown robustness. For comparison purposes, a manual analysis of these programs was also performed in order to discover which robustness properties each program satisfies or fails to satisfy. The programs under consideration, with the actual properties satisfied by each are listed below:

- RCA, a (stateless) ripple carry adder that satisfies stretch robustness.
- RISE, a program that receives a Boolean stream and detects the presence of rising edges, and which satisfies immediate-at-first robustness.
- SWSR, a switch with a set and reset input, which satisfies stretch robustness.
- TOGGLE, a switch with a toggle input, which does not satisfy any property for every latency function.
- SISO, a serial in serial out register, which satisfies stutter robustness.
- PIPO, a parallel in parallel out register, which satisfies stutter robustness and immediate-at-first robustness.

We shall now discuss the application of the static and dynamic analysis theorems to the programs in question. To illustrate how the static analysis theorems can be employed to reason about a program, we will consider their use to prove that the SISO register program is stutter robust.

Example 1: Since the SISO program has 4 equations, we first break it down into four separate primitive programs SISO_1, SISO_2, SISO_3 and SISO_4 as shown in Figure 7, where $\text{SISO}_j = \langle\{i_j, i_{j+1}\}, \{i_j\}, \{i_{j+1}\}, \{i_{j+1} = true\ fby\ i_j\}\rangle$

It is clear that each such program is an fby primitive program, and that these primitive programs can be composed through fully connected composition to obtain the program

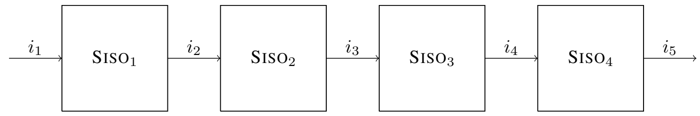


Figure 7. SISO program broken into primitive programs

SISO. This can be done by starting from SISO_1 and sequentially composing the programs SISO_2, SISO_3 and SISO_4. Since SISO can be built from stutter robust primitives and through stutter robustness preserving compositions, we can conclude that it is stutter robust.

Table I illustrates the results that can be obtained in a similar manner through the static analysis of the programs in question. An entry in the table indicates whether the corresponding program can be shown to satisfy a particular robustness property or not through this technique. Within an entry, a ✓ symbol indicates that the program was found to satisfy the property. In addition, a ? symbol indicates that the static analysis yielded an inconclusive result, while a − symbol indicates that a test was unnecessary since the program was found to satisfy the stronger property of stretch robustness.

TABLE I. RESULTS OBTAINED THROUGH STATIC ANALYSIS

Property/Program	RCA	RISE	SWSR	TOGGLE	SISO	PIPO
STR	✓	?	?	?	?	?
STU	-	?	?	?	✓	?
IAF	-	?	?	?	?	✓

As one can see, the static analysis reveals that the ripple carry adder is stretch robust, that the SISO register is stutter robust and that the PIPO register is immediate-at-first robust. Static analysis thus yields results when the programs have a simple structure in terms of the interconnections between the component primitive programs.

We now illustrate how dynamic analysis can be applied by means of another example. We shall show that the Toggle Switch program TOGGLE is both immediate-at-first robust as well as fast-enough robust for the latency function $\lambda(t) = 2$.

Example 2: Starting from the TOGGLE program, we first obtain the automaton representation of the program by using the construction outlined in Definition 1. This yields the automaton depicted earlier in Figure 1. By observing the structure of the automaton, we note that from any state, taking 2 transitions with the same input tuple returns the automaton to the same state. The program thus satisfies the properties in question through the use of Theorem 18.

Table II summarises the results obtained through the dynamic analysis of the programs under consideration. In addition to the earlier conventions, an $\times$ symbol indicates that the program was found not to satisfy the property in question, while a ✓ symbol with subscript $c = 2$, indicates that the program has been proven to satisfy the property for the latency function $\lambda(t) = 2$ through the use of a latency dependent condition. In practice, BDD techniques were used to evaluate the conditions, and the evaluation was instantaneous for the programs in question.

TABLE II. RESULTS OBTAINED THROUGH DYNAMIC ANALYSIS

Property/Program	RCA	RISE	SWSR	TOGGLE	SISO	PIPO
STR	✓	×	✓	×	×	×
STU	-	×	-	×	?	?
IAF	-	✓	-	$\checkmark_{c=2}$	?	✓
FE	-	?	-	$\checkmark_{c=2}$	?	?

Dynamic analysis enlarges the scope of automatically derived robustness results to programs that have more complex structures. In particular, this approach gives the following results.

- *Stretch Robustness:* The ripple carry adder is shown to be stretch robust, a fact also discovered through the static analysis. On the other hand, the switch with set and reset is shown to be stretch robust through condition 3, while all other programs fail to be stretch robust by condition 2. The latter two results are an improvement over the analysis conducted in [1].
- *Stutter Robustness:* The rising edge and toggle switch programs are shown not to be stutter robust, something which is not possible to discover through the static analysis, since no conditions for the failure of a property are considered.
- *Immediate-At-First Robustness:* The PIPO register and the rising edge program are shown to be immediate-at-first robust. While the former fact was discovered by the static analysis, the latter one was not.
- *Latency Dependent Conditions:* The toggle switch program is shown to satisfy immediate-at-first and fast-enough robustness for the specific latency function $\lambda(t) = 2$.

While not all of the properties satisfied by the programs have been discovered through the automated analysis, the combination of static and dynamic analysis has revealed many details about the robustness of the programs in question. The conditions for stretch robustness have in particular been effective at classifying the programs considered. The number of programs proven to be immediate-at-first robust also indicates that the condition which detects it might be applicable for some interesting set of programs. On the other hand, the results obtained using the latency dependent conditions are encouraging as they indicate the possibility of satisfying a property under a particular slowdown scenario even though the program might not satisfy it in general. One can also note that the two approaches can be complementary; in particular while dynamic analysis allows discovery of programs that are not stutter robust, static analysis allows reasoning about those that satisfy this property.

VII. RELATED WORK

The discrete theory of slowdown considers the effect of slowing down all the input streams of a stream processing program by the same amount through the addition of stutter. There are various other models of slowdown that can be found in the literature. The theory of latency insensitive design [6] allows streams to slow down through the addition of explicit stall moves into those streams. In reaction to performing a stall move on an input stream, a program reacts by performing a procrastination effect, that is by inserting additional stall moves in its other streams to ensure that causality between the events of a program is preserved. A program is said to be patient if it knows how to perform a procrastination effect in response to any possible stall move. In other words, the program is always able to delay its operation in response to slow input without breaking. Patience is thus a form of robustness to delays in the process' streams, but which, unlike our properties, does not dictate the exact form that this robustness should take.

In the theory of polychronous processes [5], used to give a semantics to the synchronous language Signal [4], streams do not have to take a value at every time instant. Given a particular program behaviour, consisting of the input and output streams of a program, the operations of stretching and relaxation can be used to obtain a slower program behaviour. Stretching remaps the time instants at which the values occur on each stream, preserving the order of values in each stream, and the simultaneity of values between different streams. The stretching operation stretches all the streams by exactly the same amount and is similar to how a stretch robust program would behave when its inputs are slowed down. On the other hand, when relaxation slows a behaviour, it only guarantees that the order of values within each stream is preserved. The notion of relaxation that arises when all input streams are slowed down by one amount, and all output streams by another amount, is similar to how to a stutter robust program would behave under input slowdown. Signal guarantees that all its programs are stretch closed (a property analogous to stretch robustness), but this is only possible because no additional values are ever inserted as a result of slowing down a stream.

Reasoning about slowdown and speedup for continuous time behaviour has been investigated in [3]. The behaviour of a program can be slowed down by stretching these real-time signals through time by using the concept of time transforms. The concept of a latency function can be seen as a discrete time version of a time transform. When one slows a behaviour through a time transform, all streams are slowed down exactly by the same amount. This manner of slowing down a behaviour corresponds to how one would expect a stretch robust system to react in our discrete theory.

Stutter invariance for Linear Temporal Logic (LTL) properties has been investigated in [7], in which a stuttered path slows down all inputs and outputs of a program by the same amount. Stutter invariant properties are ones which, if they are satisfied by a program, then they are also satisfied by all stutterings of its behaviour. If a Lustre program is stretch robust, then its inputs can be safely slowed down without the risk of breaking the constraint imposed by a stutter invariant property on the program.

The theory of stability [10] considers programs whose outputs fluctuate when their inputs are kept constant. Programs that do not exhibit such a phenomenon are said to be *stable*; when the inputs of these programs are unchanged, the outputs will converge to stable values after a finite period of time. The concept of stability relates to the concept of fast-enough robustness. An input that has stopped changing is similar to an input which is stuttering when considered over some finite horizon of time. While the theory of stability requires the output of a system to eventually converge to some particular value, fast-enough robustness requires an output to converge to an expected value before the sequence of repetitions ends.

Instead of checking whether a system exhibits certain classes of behaviour when an environment changes, it is possible to check whether a system degrades gracefully when the environment misbehaves. In [11], the authors consider a robustness approach in response to environments that fail to obey the assumptions made during system design. A system is said to be robust, if a small number of violations of the environment assumptions causes only a small number of violations of the system specification.

It is also possible to use a probabilistic approach to understand how changes in the environment are propagated through the system's components, and how the behaviour of these components under changed or missing input contributes to cause unacceptable system wide behaviour [12]. From our perspective, the general approach is interesting because it can help to isolate those components that misbehave under slow input, causing a complex system to fail.

VIII. Conclusions

Since input stutter can arise in various situations, especially in systems that finely sample input, it is crucial that such systems do not change their behaviour as such transformations on their input occur. In this paper, we have identified a number of different levels of robustness with respect to slowdown which one may require, and presented sound checks using static analysis of the code or using symbolic verification techniques over the system's behaviour.

One important observation regards the strong (and highly compositional) property of stretch robustness. This property is found in various guises in slowdown models that are simpler than our own. These models are often found to either assume that programs will be stateless, or else will define slow input in a way that does not bear on program state. In our model, stutter is able to interact with the internal state of the programs under consideration resulting in outputs with more complex relationships to those generated under the unslowed input. Nonetheless it was found that even under these conditions, there are non-trivial stateful programs that remain stretch robust to input slowdown.

Nonetheless, the restrictions that are imposed by stretch robustness in the complex slowdown scenario considered are quite strong, and consequently it is useful to possess weaker properties for those situations in which stretch robustness does not apply. Weaker robustness properties are however not as compositional as stretch robustness, making a static analysis approach less powerful than it would be with the stronger property. On the other hand, we have seen that dynamic analysis allows for the analysis of programs on a global level, allowing greater effectiveness in checking all kinds of robustness property - albeit at an increased computational cost. The two approaches can, however, be combined, allowing for the analysis of more complex programs.

One major restriction of our results is that we assume that all the inputs of the system are slowed down by the same amount. In practice, this may be too strong a restriction, for instance with some nodes using a combination of external inputs and streams coming from other nodes and which may have been slowed down further.

Another restriction is that we limit our analysis techniques to Boolean Lustre programs or circuits. The static analysis results, depending only on abstract properties of primitive programs and the way they are interconnected, can be applied to programs over non-finite types. On the other hand, all dynamic analysis results can only be applied on Boolean Lustre programs. In the future, we plan to relax this constraint by using control graph analysis techniques to programs with numeric values, using approaches similar to [13].

References

[1] I. Bondin and G. Pace, "Static and dynamic analysis for robustness under slowdown," Proc. 4th International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking (COMPUTATION TOOLS '13), IARIA, May 2013, pp. 14–22.

[2] P. Caspi, D. Pilaud, N. Halbwachs, and J. Plaice, "Lustre: a declarative language for programming synchronous systems," Proc. 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages (POPL '87), ACM, Jan. 1987, pp. 178–188.

[3] C. Colombo, G. Pace, and G. Schneider, "Safe runtime verification of real-time properties," Proc. 7th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS '09), Springer Verlag, Sep. 2009, pp. 103–111.

[4] A. Gamatié, Designing embedded systems with the SIGNAL programming language - synchronous reactive specification. Springer, 2010.

[5] P. Le Guernic, J.-P. Talpin, and J.-C. Le Lann, "Polychrony for system design," Journal of Circuits, Systems and Computers, vol. 12(3), Jun. 2003, pp. 261–304.

[6] L. Carloni, K. Mcmillan, and A. Sangiovanni-Vincentelli, "Theory of latency insensitive design," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 20(9), Sep. 2001, pp. 1059–1076.

[7] D. Peled and T. Wilke, "Stutter-invariant temporal properties are expressible without the next-time operator," Information Processing Letters, vol. 63(5), Sep. 1997 pp. 243–246.

[8] N. Halbwachs, P. Raymond, and C. Ratel, "Generating efficient code from data-flow programs," Proc. 3rd International Symposium on Programming Language Implementation and Logic Programming (PLILP '91), LNCS 528, Springer Verlag, Aug. 1991, pp. 207–218.

[9] S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDL Design, second edition. McGraw-Hill Higher Education, 2005, pp. 522–531.

[10] N. Halbwachs, J.-F. Héry, J.-C. Laleuf, and X. Nicollin, "Stability of discrete sampled systems," Proc. 6th International Symposium on Formal Techniques in Real-Time and Fault-Tollerant Systems (FTRTFT '00), Springer Verlag, London, Sep. 2000, pp. 1–11.

[11] R. Bloem, K. Greimel, T. Henzinger, and B. Jobstmann, "Synthesizing robust systems," Proc. Formal Methods in Computer-Aided Design (FMCAD '09), IEEE Computer Society, Nov. 2009, pp. 85–92.

[12] X. Ge, R. Paige, and J. McDermid, "Probabilistic failure propagation and transformation analysis," Proc. 28th International Conference on Computer Safety, Reliability and Security (SAFECOMP '09), LNCS 5775, Springer Verlag, Sep. 2009, pp. 215–228.

[13] B. Jeannet, N. Halbwachs, and P. Raymond, "Dynamic partitioning in analyses of numerical properties," Proc. Static Analysis Symposium (SAS'99), LNCS 1694, Springer Verlag, Sep. 1999, pp. 39–50.

Human Cooperation Improvement Using Autonomous Skill Agents

Olivier Chator
Conseil Général de la Gironde
IMS Laboratory, CNRS, IPB
Université de Bordeaux, France
o.chator@cg33.fr

Pierre-Alexandre Favier and Jean-Marc Salotti
Ecole Nationale Supérieure de Cognitique
IMS Laboratory, CNRS, IPB
Université de Bordeaux, France
pierre-alexandre.favier@ensc.fr, jean-marc.salotti@ensc.fr

***Abstract*— A local authority, the "Conseil Général de la Gironde" in France, manages various projects in different fields, like sustainable development, and coordinates public and private partners' actions. The observation shows that each of them has only a partial vision of others' skills and know-hows. Generally speaking, everywhere where human collaboration is needed, sharing skills is one of the problems identified. This work addresses these difficulties using a learning and collaborative multi-agent system to enhance skill sharing and management. One of the main innovations here is that skills are represented as autonomous agents, and not just as capabilities, as is usually the case.**

***Keywords*-Multi-agent systems; sustainable development; skills; governance**

I. Introduction

A local authority, the "Conseil Général de la Gironde" (CG33) is responsible for public actions for 1.5 million inhabitants. Numerous domains are concerned: school transportation, management of middle schools, tourism development, solidarity, integration and support for elderly people. One of the CG33 missions is to define policies and practices for the Sustainable Development (SD) of the department (a territorial division lower than regions). For example, the objective could be to transform a neighborhood into an eco-district [1, 2]. Experience shows that this type of project is very complex and requires the collaboration of many public and private actors under the supervision and management of a "project supervisor" (PS), for instance an architectural firm. Each actor has only a partial knowledge of the capabilities of the other and some information is sometimes lacking, but the PS has to take decisions anyway. In addition, the objective is often to minimize the costs and to obtain energy or ecological labels, which typically are antagonist objectives. For the PS, it is often difficult to understand the impact of each parameter. The preferred option is usually the one that is better understood, which comes at the expense of other options because there was insufficient knowledge on their impact, cost, and implementation. In order to help the actors, and especially the PS, to find the best partners, the CG33 decided to build a database of skills and actors [1]. For example, it should be possible for a PS who wants to renovate some buildings to identify skills and actors in various domains such as thermal insulation, thermal simulation, air tightness, and installation of different types of photovoltaic panels on the roof. In turn, the partner who has expertise in thermal insulation may require the help of another partner who is specialized in the use of specific insulation materials. Thus, the challenge is here to allow each stakeholder of an SD project to share and learn more about the expertise and know-how of the others. In general, whatever is the field of activities (e.g., building the best sport team as possible), everywhere where human collaboration is needed, the problem is the same. Therefore, to facilitate the increase of skills for each actor over time and to stimulate their cooperation, building an efficient system of skill sharing is the key.

A traditional approach could be to build a simple database with a direct link between actors and skills. However, considering the central role of skills and the needs for constant evolution and modifications of the data, a research project has been carried out in our laboratory to find and implement a better solution. It is suggested here that a multi-agent system (MAS) is more appropriate. In our proposal, the innovative key concept is to consider each skill as a "full agent", and not only as an agent's ability as is usually the case [3, 4, 5]. Having their own learning model and their own life cycle, our skill agents are autonomous, cognitive, and they interact with human actors to stimulate and improve cooperation. Finally, this work offers a proposal to improve the management of skills, to make them more efficient, during projects, and over time. If a user wants to address a new goal, how should he define the project and what are the skills needed to implement it? How can he get benefits from past projects?

The model is described in Section II. Some results are shown in Section III and Section IV concludes this document.

II. Model

A. The Issue

Let us introduce the issue with an example of an SD project that aims to "transform a neighborhood into an eco-district". Let us assume that a PS has to build a HQE building. HQE stands for "Haute Qualité Environnementale" and is a standard for green building in France [6]. For this kind of project, the PS needs:

- A definition of the goal to achieve (the objective)
- Skills such as "integrating insulation materials" in order to meet the HQE objectives.

- Actors such as private building companies to implement the skills.

Using our "SD skill sharing" system, the PS should be able to identify a list of possible partners. Intuitively, we might think that this list could be simply sorted according to the most experienced partners for the given task. However, other criteria than just experience should be taken into account: price, quality, duration, localization, expertise with specific materials, etc. The system may suggest a partner according to this list of criteria. In addition, it also has to select different companies over time. The problem is to determine a good strategy in order to achieve that goal throughout a project.

Going further, it is also interesting to get the benefits of past experiences on similar projects over time, to build new ones. Let us illustrate with the objective: "*I want to put photovoltaic panels on the roof of my house*". I have to find the skills required for this new project. Taking my needs and experience of past projects into account, there are two possibilities:

- I find a past project that reflects exactly what I want to do. Thus, what I need is to find a way to retrieve all the skills of this project, and proceed to my new project creation using this list.
- I find a past project, but it is not exactly what I want to do. Thus, what I need is to find a way to retrieve some of the skills of this project, and proceed to my new project creation using this restricted list.

The two points above are efficient if the user finds projects that already contain all or some of the skills he needs to build the new one. However, this is not always the case. Skills may be scattered throughout various projects. Thus, the point is to find a way to answer a limited expression of needs at specification level. For example, if we consider that in the system we already have the past two projects: "*wind turbine implementation*" and "*hydraulic micro power implementation*". They both belong to the same "domain": "*new means of energy production*". If the user wants to build a new project, in the same domain (e.g., the implementation of solar panels), an interesting idea is to look for skills used in all the past projects of this domain. Thus, the system will suggest integrating the skills that were fluently used across all projects in the domain.

It is possible to generalize from this example. Each professional sector of activity has procedures and processes, each of them used to reach specific goals or objectives. Each objective may be implemented through some projects and skills. Generally speaking, it may be difficult to make processes evolve according to environmental constraints. When successful, the knowledge and know-how that have been used should be capitalized on for possible use in further projects. Finally, the main problem is to find a way to improve the management of skills to make projects more efficient over time. To do this, it is possible to build new projects, working on past projects or objective domains. Thus, the goal of this work is to improve new project definitions dynamically and to identify all the skills needed to make them a success.

B. Defining a Skill

A skill is the ability to exploit some knowledge and know-how in order to solve a class of problems. It is different from a competency, which is generally accepted as a set of behaviors or actions needed to be performed successfully within a particular context [4]. In this study, for the sake of simplicity, it is assumed that a skill is a sum of elementary competencies (ECs).

The main specifications of our application are to store information about the skills of possible participants in SD projects and to suggest interesting partners for a given skill. An important issue is to make the link between observations (e.g., "partner A has been assigned the role of task 1 and 2 in project X and has succeeded in implementing solutions") and skills, which do not correspond to the names of the task. Let us present an example:

Integrating glass wool for the insulation of northern walls in a specific building in a given project is related to the skill "integrating insulation materials". However, integrating isolated wood panels under roofs might be very different from integrating glass wool in walls and the best expert for the first task might not be the best one for the second. The skills might be differentiated by small details, but, for the proposed application, it would be irrelevant. It is expected that the users of the application will ask general questions such as "who has skills in insulation materials?". The key problem is to find the appropriate level of detail for each skill and to make the difference between an elementary competency that belongs to a skill and the skill itself. Then, assuming that a skill is defined at the right level and includes a list of possible elementary competencies, the question is to determine how each of them participates in the definition of the skill. For instance, for the skill "integrating insulation materials", how important it is to have the know-how for isolated wood panels? In other words, there should be an associated weight for each elementary competency and there should be a mechanism for learning them integrated in the skill agent

According to the needs of the project, a skill can be created at any moment, its definition (the list of elementary competencies) may evolve, it can eventually be split into different skills and it might even be removed. Such constraints cannot easily be handled in a standard database in which the actors and their skills would be stored. Because of the central role of the skills, it is suggested in this paper that the skills be considered as agents of a multi-agent system. However, in most applications, agents are associated with models of actors in the real world, the skills defining the behavioral rules [2, 7, 8]. The problem is that the skills have their own dynamics and are rather independent from the actors. The skills should be agents with their own lives. In addition, if the skills and the actors are distinguished, it is difficult to define actors as other agents of the system. In cognitive science, the embodiment of mind is often considered a requirement to obtain an effective agent [9, 10, 11, 12]. Skills alone have no perception, no motivation and no means to perform an action and change their environment. Nevertheless, it is possible to define these elements artificially. Intuitively, a skill can be motivated by the improvement of its own definition, e.g., a weighted list of elementary competencies and the clarification of its relationships with the other skills. The user of the system has another motivation: he wants to find a partner for his project. The system should provide some criteria and suggest an actor for the required skill. The user makes his choice, then the work is carried out (embodiment of the skill) and an evaluation of the realization is performed. The key idea is to consider that a criterion is no more than an abstraction of a hidden list of elementary competencies. For example, the duration of a work is not a competency. However, implicitly, it is closely related to the ability to work fast, which is an elementary competency of the skill. Therefore, the skill can exploit the definition of the criteria, which evolve according to the evolution of the projects, to characterize its definition. Another issue concerns the links between the skills. Different skills may have several elementary competencies in common. If no actor is found for a given skill, an interesting idea is to make suggestions with actors associated with the skills that are closely related.

In addition, a database is still required for the storage of past observations (e.g., Actor A has been involved in project X for the embodiment of skill S with an evaluation of a list of criteria $C_1 ... C_n$).

C. Definitions and key concepts

1) Main concepts

We can use another concrete example to illustrate those concepts: "*To make energy savings, I want to put better insulation into the walls of my flat*". From this example, we may define four concepts in our proposal.

a) Environment

An environment is viewed as a professional sector of activity. In our example, "*sustainable building sector*" is the environment in which the user request occurs. This is the highest level of abstraction and it is related to the professional sector of activity.

b) Objective domain

An objective domain is a group of objectives, concerned by the same theme of activity. In our example, "*thermal insulation improvement*" is the objective domain in which the user request occurs. Another domain could be "*air tightness improvement*". The idea is to position objectives within one or several objective domains.

c) Objective

An objective is a simple textual description of a goal to be reached (NB: the term "goal" would have been more appropriate, but "objective" was chosen from the start for convenient reasons and links with the French language). In our example, the objective is to "*put better insulation into the walls*". This objective is part of the "*thermal insulation improvement domain*". Considering this simple question from the user, no constraint about the materials or skills used to reach the goal is expressed. Another objective could be "*improve air tightness in my flat*". In our implementation, an *Objective* agent is available. To transpose an objective into "real life", it is mandatory to define a project first.

d) Projects

A project is defined by an objective, a start date, an end date, resources (like human actors) and processes to schedule the list of skills to be used. In our example, a project defined by the "*integration of glass wool into walls*" is proposed. It will start next week, will stop in 15 days, and requires several skills. Finally, the project is implemented and evaluated at the end using ECs of each skill.

2) Types of Objectives

For a user, the problem is to reach a goal. It is usually defined by a simple assertion like "*I want to do something*". In order to reach the goal, the user will define a new project in our system. He does not often have knowledge of all the skills that have to be used in the project. As it is difficult to give a unique answer, depending on the user request at time T, two approaches are proposed for building new projects. The first one is based on completed projects and their objectives. The second consists of building the new project using only concerns about the objective's domain, and possibly the environment. This case occurs when the user wants to do actions in a particular domain of activity, but does not know exactly what to do.

D. Skill Agents: theoretical proposal

1) Introduction

Generally, an agent has behaviors. Each of them could be presented as shown in Figure 1.

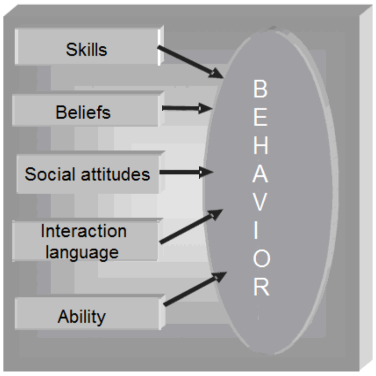


Figure 1. Agent's behavioural characteristics

It is assumed here that a skill is unique and can be implemented as an agent in a multi-agent system. It has resources (a list of physical actors) and its own life cycle. It can be created, can evolve and can eventually be removed when not used anymore or when replaced by another skill agent. Skills agents fit into a multi-agent system, where the environment is defined by the interactions with the users. They are cognitive, non-conversational and non-dialogic [3, 5]. They never directly communicate with human users. They react and evolve according to information modifications and requests from the user via a *WebRequester* agent. An important feature is their ability to learn how to define themselves and how they are linked to the other skills.

2) Definition

Skill agents are defined by three main features: perception, internal attributes and actions:

- **Perception**: Skill agents are listening to information broadcast by the system after interaction with the users. It can be, for instance, an update after external observations (e.g., a new project is started or the result of work for a given project is inserted in the database) or a request is sent by another agent within the MAS.
- **Internal attributes**: A skill agent is determined by the list of elementary competencies that defines the skill, a creation date (appearance in the MAS), a domain(s) of activity and a specific "*age*" (see below). It also has a list of behavioral rules, expressed in XML format with a specific grammar (see below).
- **Actions**: If there is an update of an external observation that is linked to the skill, the agent updates its database and its weights according to a learning rule. It provides an answer to the *WebRequester* agent (which, in turn, informs the user) according to a strategy defined by behavioral rules. Each skill agent has the ability to establish links with other skill agents in the MAS. This last action is based on its environment analysis, automatic (or not, if specifically requested by the user), and defined in its behavioral pattern. More importantly, skill agents are proactive. When a user wants to create a new project (typically a new objective action), an *Objective Agent* receives those requests. Then, it sends a broadcast to inform all skill agents. Each of them determines if it is a candidate for participation (or not) in the new project. The decision is defined through the computation of what we call a "proximity coefficient" (see below). At the end of treatments, the *Objective Agent* returns a list of candidates to the user for (in)validation according to the project needs.

3) Life cycle

Skill agents have their own life cycle, divided into 3 "ages", according to their specific levels of autonomy.

1. **Childhood**: The skill agent runs in a "learning" mode. During this age, the aim is to make the agent "grow". When it is created, the first step is to assign to it a list of criteria (elementary competencies) for future evaluations. The second step is to associate a list of actors. At initialization time, there is no evaluation in the database because the agent has not been used yet. Thus, if a user looks for an actor (like a rugby player) for this skill, the agent is not able to make relevant suggestions (childhood). It simply returns a list of potential actors ranked according to the number of times they have been involved in realizations (the most experienced at the end). Once the result of the work is available, the user evaluates the criteria associated with the skill and the data are stored in the database. At this age, the skill agent does not really communicate with other agents and uses only basic behavioral rules. It grows until it reaches a threshold of evaluations (e.g., after 3 concretizations across various projects) in the database. When this threshold is reached, its age automatically grows to the next one.
2. **Teenage**: When a user looks for an actor with this skill, the agent computes a list of candidates, exploiting the previous results (past experiences) and the current user criteria weights values. If the user does not specify any weight for criteria, default values are used (e.g., weight=1). A list of potential actors is then obtained after a dynamic computation of criteria weights (see subsection D.2). At this age, the autonomy of the agent is rather limited. It communicates and tries to build relationships with other agents (see subsection D.3). Whatever the action performed, a human validation is requested and stored in the skill agent memories table in the database. The skill agent grows until it has a threshold of evaluations (e.g., after 10 concretizations across various projects) in the database. When this threshold is reached, its age automatically grows to the next one.
3. **Mature:** The skill agent is able to make direct and relevant suggestions to the user as soon as a project is created. A list of skills is proposed with the possible actors for each of them. Obviously, the user can still make modifications but he can save a lot of time if the choices correspond to his needs. At this age, the agent has a good knowledge of its relationships with the other agents. Using the system parameters, it is possible to cancel the validation of the system choices by humans. For example, in our SD skill sharing system, we could imagine characterizing a new project (e.g., building a new middle school) without knowing what the skills needed are, nor the actors who are able to implement them. In this situation, at the mature age, the system will automatically choose the skills (using the links between them) and will affect their concretization to actors. Thus, the skill agent is

fully autonomous, makes its own decisions, and does not need human validation.

Remark: the thresholds for age transitions have been set empirically. The objective is to grow as rapidly as possible. It is a trade-off between giving a help to the user as soon as possible, and reaching a high level of expertise.

E. Learning mechanisms

1) Introduction

In the literature, we can find various types of learning mechanisms for agents: the Markov Decision Process [13] with reinforcement learning [14], the theory of games (matrix games [15] and stochastic games [16]), the Bayesian networks [17], the Case-Based Learning (CBL) methods [18], and so on. Our skill agents evolve according to human actions on the system (requests, selections, validations, and evaluations). In our context, the CBL methods seems more adapted because they are based on valuated and memorized iterations of concrete experiments, they integrate validations of human on the system decisions, and allow (for a given environment) specific and gradual adjustments over time. However, existing CBL algorithms [19, 20, 21, 22, 23, 24, 25] are not entirely appropriate to our problem because the objective is not to find a similar case in the knowledge database. Selected actors will have the skill anyway. It is rather to make a choice among several possible actors according to a global skill sharing policy.

Therefore, synthetically, learning mechanisms for a given skill agent, are based on two main points: actors' selection mechanism and the building of links with other skill agents.

2) Actor selection

When the user needs to find an actor for a given skill (generally inside a project), he asks the system for suggestions. We have already seen that a skill is defined by a weighted list of criteria or elementary competencies.

The first step is to give this list with undetermined (or default) weights to the user. For instance, in an SD activity domain, if the user wants to retrieve an actor for the "thermal insulation" skill, the system asks the user to define the weight associated with each criterion: "wall insulation", "roof insulation", "wood based materials", "diagnostic", "price", "duration", etc. This information is used to update the definition of the skill.

Assuming the user gives specific values to the weights of each criterion, the second step consists in computing the new weights. This is done using those new values plus the old validated computed weights values. "*Validated*" means approved by the human user in past experiments. Let **k** be a criterion and $\mathbf{W_{k(t)}}$ the weight associated to it for request number **t.** For **t=1**, the average weight is set to $\mathbf{W_{k(1)}}$ which is the weight given by the user. For **t > 1** the new average weight is computed using equation (1).

$$\overline{W}_{k(t+1)} = \frac{(t \times \overline{W}_{k(t)}) + W_{k(t+1)}}{t+1} \tag{1}$$

The third step is to build the list of actors, evaluating them. The proposal is to use the new computed weights of each criterion, and the evaluation results of previous realizations (experiments) of the skill. Let **f** be the number of times where an actor **a** has concretized the skill in past projects. Firstly, we compute a partial value using the old "after work" evaluations (equation (2)):

$$\overline{E}_{k(a,t)} = \frac{\sum_{j=1}^{f} E_{k(a,j)}}{f} \tag{2}$$

Secondly, the "actor evaluation" is done with equation (3):

$$Eval(a) = \sum_{k=1}^{t} \overline{W}_{k(t)} \times \overline{E}_{k(a,t)} \tag{3}$$

Please note: if the user has given specific values for the criteria weights, we will have $\overline{W}_{k(t+1)}$ instead of $\overline{W}_{k(t)}$.

In order to give a chance to each actor, we propose to alternate between the performance policy (linked to evaluations) and the skill sharing policy (a random selection process), with a probability of 0.5. Thus, an actor with systematic lower evaluations is not penalized. Additionally, in any case, the user can still select an actor who is not at the top of the list. Considering the skill agent life cycle (as seen in Section III), a human validation of the system choice is needed when the skill agent is in childhood or teenage age. When it is mature, the validation is considered as implicit.

The validation process consists in updating the skill agent memories, typically a record in a table, where the current computed weight $\overline{W}_{k(t+1)}$ of each criterion was stored. When the choice is approved, by the user or automatically at mature age, those weights become validated $\overline{W}_{k(t)}$. Being now part of what we called "*old validated computed weights values*", they will be used for the next $\overline{W}_{k(t+1)}$ computation. Such a method ensures a learning activity, reflects "real life" situations, and is really close to what is done in Case-Based learning systems.

3) Agent links

For some projects, a user who is not experienced may not necessarily know all the skills that are required for the realization of a project. Concrete example: "I am working in a local authority, in a little town, and would like to organize a public event to stimulate sustainable behaviors around waste sorting in my citizens". In "real life", this is a case of use of our SD skill sharing system in Gironde. To solve my problem, I can ask the system to help me and suggest a list of skills in order to succeed. The problem is the same if I am the coach of a rugby team: "I would like to organize some specific training for my players at such a position. What do I have to do, who can help me?"

In a MAS or agent's point of view, in order to provide an efficient answer, it is possible to exploit the links that can be found among the skill agents that are at least in the teenage age. For a given skill agent S, the proposal is to build links with other skill agents by computing for each of them a "proximity coefficient" (in percentage) to S. This coefficient is built using what we call "similarities" to S that are sought in the MAS environment. Similarities are found by searching for elements that S and the other skills have in common, using dynamic requests generation in the database.

Let us make an assumption: a similarity between two skills is defined by the number of common descriptors among the skill internal attributes, the skill domain, the elementary competencies, the past projects, the evaluation of the skill in previous projects, the project domains, and finally the types of project domains. This information can automatically be obtained using the database table tree. Currently, the database structure analysis shows that, in this tree, the lower the depth of a table is, the more significant the similarity also is (see Figure 2). To find similarities, the S skill agent analyses the database table tree (the MAS environment), dynamically and recursively.

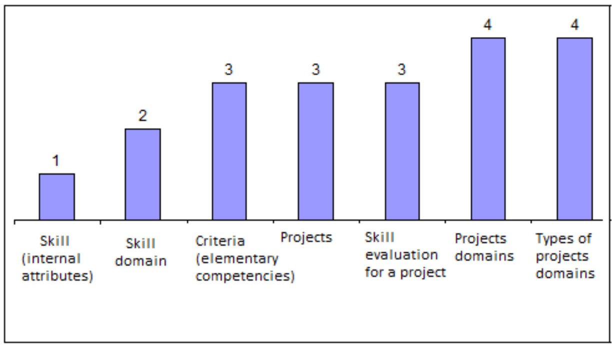


Figure 2. Similarities table depths

Let us assume that the maximum depth of the table tree is **D, which means that each skill can be reached from another skill after D links.** D=4 in the example of Figure 2, at the beginning of the tree exploration, the current table depth is **d=1**. The analysis starts with the global skill agents table.

Step 1: for each field of a current table, an SQL request is dynamically generated to find out the other skills sharing the same field value.

Step 2: if a skill S_i is found, the first time, its current proximity coefficient is $\mathbf{P_{Si} = 0}$.

Step 3: considering **n** as the number of common field values into the current table, P_{Si} is updated using equation (4).

$$P_{Si} = P_{Si} + (n \times (D - d)) \tag{4}$$

Step 4: Foreign keys and joined tables of the current table are used to define the next nodes of the tree. d is incremented: **d=d+1**.

Step 5: if **d** is not greater than D, go to step 3

Step 6: finally, all P_{Si} are normalized using equation (5):

$$P_{Si} = \frac{P_{Si}}{P_{Si\max}} \times 100 \tag{5}$$

The higher the proximity coefficient, the more skills are considered as potentially linked. A threshold is applied on Ps (e.g., 65% by default), and a list of potential linked agents is returned to the user for validation. An important point to know: validating the links implies the validation of a dedicated memory and an automated evolution of S agent behavioral rules. Thus, there is here again a real learning mechanism because of those evolutions arising from human validation. Reminder: when a skill agent is in the mature age, no human validation is needed and the agent is completely autonomous, generating links and updating its own behavioral rules.

F. Improving skill management using objectives

1) Introduction

"I want to improve the energy balance of my house by installing photovoltaic panels on the roof". "I want to win the rugby match next Sunday". Generalizing, the question is always to find the best answer to questions like "I want to do something to achieve something". Then the project comes into place. Goals and objectives are statements that describe what the project will accomplish. Each project is defined (structured) by a set of resources and processes according to a specific schedule. The processes are based on a set of required skills. Importantly, the definition of the project may evolve according to environmental conditions or past experiences, meaning that the list of skills required to reach the goal may not always be the same. Let us consider the goal of "building a house". Even if the process is almost the same, different construction materials (like cinderblock or brick) may be used. This implies different skills for each building project. Considering past experience in the domain, a dynamic dimension is observed for each project over time.

Then, the point is how to improve the management of skills to make them more efficient, throughout projects and over time. If a user wants to address a new goal, how should he define the project and what are the skills needed to implement it? How can one get benefits from past projects?

In order to solve the problem, work on the goals and skills of past projects is suggested.

2) Building new projects from past projects

Two cases are available to build the new project, by duplicating or customizing existing ones. From our previous example, we will suppose that the user finds the project "*integration of glass wool in walls*". He may then:

- Think that it is exactly his objective. Thus, he will duplicate this project and all its skills, without creating a new one, changing only contextual information like the start date for example.
- Think that it is not exactly his objective, but is very close. He decides to create a new project and customizes the list of skills associated with the existing project. A new objective is thus created with a new list of skills built from a subset of the previous one.

Finally, in both cases, a new project instance is generated from the objective. In our implementation, those operations are done throughout our unique *Objective* agent into the MAS. As those actions are based on historical data, no communication with skill agents themselves is needed. Let us assume that a new project has to be built according to the user needs. There is sometimes a limited expression of needs at the specification level and the objective might be met for the first time. The user would probably not know how to address the problem and how to exploit past projects. There is nevertheless a solution to help the user. The idea is to determine all the skills involved in past projects in the same objective domain, or eventually in the same environment. In our system, a skill "***wants to be involved***" in the new project according to a degree of involvement in past projects (see below). In contrast to what we have seen before, this method is not limited to an exchange with the *Objective* agent. In cognitive science, effective agents are obtained by the embodiment of mind [11, 12]. If a skill alone has no perception, no motivation and no means to perform an action and change its environment, it is always possible to define it artificially. Several types of motivations have been integrated in skill agents: contribution to new projects, determining the list of elementary competencies that define themselves, and determining their relationships with the other skills. Let us develop an example showing the motivation of being involved in new projects. A user defines a project in the "*thermal insulation improvement*" objective domain. No more detail is forwarded to the system. "*Thermal insulation improvement*" is an objective domain and is part of the environment "*sustainable building sector*". The answer from our system is defined by the following process:

- The *Objective agent* receives the user request.
- The *Objective* agent sends (broadcasts) the request to all skill agents within the MAS.
- Each skill agent computes a "*relevance coefficient*" according to the request content, and returns an answer to the Objective agent (see next paragraph).
- The *Objective* agent consolidates all answers from skill agents, and returns the list of candidates to the user

Each skill agent is autonomous and decides if it wants to contribute to the new project (or not). The key point is the computation of the relevance coefficient. Currently, this is the percentage of projects in which the skill has been involved in the past among all the projects of the objective domain. If it exceeds a threshold, the skill agent wants to be involved in the new project.

III. Implementation and Results

A. Implementation

1) The MAS Architecture

The model has been implemented using the JADE MAS and standard multi-agent tools [8, 9, 26, 27]; see Figure 3.

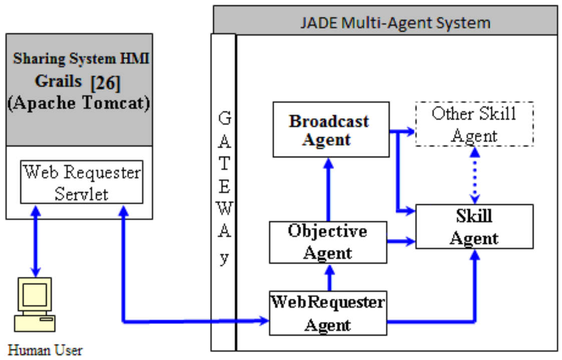


Figure 3. The MAS.

The JADE MAS has been integrated in standalone software, running into a Java Virtual Machine, and is called "SMAServer". The main components of the global architecture are:

- **User workstation**: exchange using a web browser
- **WebRequester Servlet**: This component is used for the management of the exchanges between human users and the MAS itself. It is a JADE MAS specific architecture component.
- **Gateway**: It is also a standard component of a JADE MAS, allowing dialogues among agents operating within the SMA and external programs (WebRequester Servlet) [8].
- **WebRequester Agent**: This agent is in charge of all the interactions with the human user. It forwards requests to skill agents and sends back their answers. It guarantees (FIPA compliance) that no direct exchange is possible between human users and skill agents.
- **Objective Agent**: According to Ferber's classification, the objective agent is reactive [3]. When a new project (the instance of an objective) is inserted within the system, information messages are

broadcast to all skill agents through the technical Broadcast agent one.

- **Broadcast Agent**: This is a technical (generic) agent. It receives an incoming message from a caller agent (e.g., Objective Agent), sends this message to all agents (e.g., skill agents) within the MAS and returns the answers to the caller.
- **Skill Agents**: actions have already been presented in subsection D.2. In our concrete implementation, they have sensors and effectors (as defined in standard agents theory [3]), each of them being a Java component (a class of object).

2) The user side Skill Sharing System HMI

The HMI from the user side has been implemented using the GRAILS [26] framework. There are 5 windows:

1. All skills that could be requested within the tool
2. All skills requests in progress
3. The connected user personal requests in progress
4. The skills shared (offered) by the connected user
5. The projects to which the connected user is involved. Here, the works in progress, within projects, for the connected user, are available.

TABLE I. LIST OF THE BEHAVIORAL RULES

XML Tag	Attribute	Mandatory	Comment
rules	description	X	Main tag
ruleGroup	description		Text describing the rule group
	weight		
rule	description	X	Text describing the rule
	weight		Weight of the rule in the rule group
	mandatory		Value is 1 if rule is mandatory, 0 otherwise
when	description		Text describing the condition
	sensor	X	Sensor Java class name used to verify rule condition
	params		Parameters in format name=value, separated by | character. Passed to the sensor
	result		Result variable name beginning with $
	operator	X	Logical operator used within condition expression
	table		Table name from which we try to verify condition
	field		Table field name or variable name from which we try to verify condition
	value		Value of field attribute, expressed as a regular expression
otherwise			Used if **<when>** has not been verified
do	effector	X	Effector Java class name to start if rule condition is verified
	description		Text describing the action
	params		Parameters in format name=value, separated by | character. Passed to the effector.
	result		Result variable name, beginning with $

3) Behavioral rules

The behavioral rules have been implemented in XML format with a specific grammar (hierarchy, attributes, and tags); see Table I. According to the agent links learning mechanisms, those rules may evolve over time if link creations are validated by the user. For illustration purposes, we propose below the rule ensuring that an agent will grow from childhood to teenage when 3 evaluations (e.g., after 3 concretizations across various projects) are available in the database:

```
<rule description="Growing from Youth to Teenage"
   mandatory="1" weight="1">
   <when description="Growing Conditions"
      sensor="GrowingSensor"
      params="unitaryEvaluationNumber=2"
      result="$evaluationsNumber1"
      operator="EQ" value="3">
      <do description="Grow to teenage"
         effector="Grow"
         params="from=youth|to=teenage" />
   </when>
</rule>
```

4) Skill agent memories

Each skill agent owns a dedicated memory table in which it stores the incoming parameters and the related computed decisions (see Table II).

TABLE II. MEMORY TABLE OF A SKILL

Field	Type	Comment
code	Long integer	Memory unique key code
ev_date	Date	Event date (record creation date)
evt_id	Long integer	Foreign key to event type table, describing the type of memorized event
agentid	Long integer	The current skill agent unique id
parametersin	String	Request parameters list in string format
decisionstring	String	Computed decision in string format
Humanvalidate	Boolean	Decision validated (or not) by human action
Comment	String	Free field

An example of memory record content is presented below. This result is obtained following the request for a list of linked skills to the skill with id 3192 (see Table III).

If the proposal is validated, the field *Humanvalidate* will then be set to "*true*" and a behavioral rule will be automatically generated.

5) The actor selection simulation tool

In Gironde, 61 of the local authorities are part of an "SD Network", where they share experiences and skills. They had started using and testing the system by the middle of the year 2013. The experiment concerned the management of SD projects. A preliminary study has been carried out, showing that most projects fall into 9 domains of activity.

TABLE III. MEMORY RECORD EXAMPLE

Field	Type	Comment
code	Long integer	1
ev_date	Date	2013-09-10 14:38:55.345+02
evt_id	Long integer	4
agentid	Long integer	3192
parametersin	String	controller=RDEngine;stepNumber=1;signalCode=201;action=call RdEngineForAgentLinks;fromWebInterface=true;minPercentValue=1;agentId=1508
decisionstring	String	SELECT agent.* FROM agent, agent_agt_domain, agt_domain WHERE text(agt_domain.code)=text(agent_agt_domain.agt_domain_id) AND text(agent_agt_domain.agent_domaines_lies_id)=text(agent.code) AND text(agt_domain.descriptiondomaineagt)=text('3-Diagnostic') AND agent.code<>1508
Humanvalidate	Boolean	False
Comment	String	Find agent links

These domains are: political wishes, sensitization, diagnostic, prospective, developing the strategy, elaborating the action plan, implementation of the action plan, evaluation, and continuous improvement. Skills are related to one or more domains. For instance:

- The skill "animation capability" is attached to the "political wishes" and "diagnostic" domains.
- The skill "identification and mobilization of expertise" is attached to the "prospective" domain.
- The skill "development of the sustainability report" is attached to the "continuous improvement" domain.

For skill actors selection, in order to obtain immediate results, a simulator was built to verify the theoretical proposal. It is based on elementary competencies weight computation. The simulation phase lies in a call to a unique skill agent, with the aim of observing its behavioral evolution over time. Each simulation applies random values to weights to each elementary competency (4 in total) of this skill. During the simulation phase, 100, 200 and 300 requests and validations were done.

B. Results

1) A real life experiment at CG33

It is observed that the skills evolve in a "real life" SD Skill Sharing System, at CG33, and they provide answers. The following results are presented in this context. The positive point is that the skills provide valuable information to the actors who have poor understanding of the elementary competencies. The drawback, however, is that the initialization of the system is labor intensive. The first definition of the skills requires strong expertise in the domain. The updates can be done at any time, but it takes a long time to collaborate with experts in order to capitalize their knowledge and insert relevant skills and elementary competencies into the system. Therefore, it is difficult at the moment to conclude about the efficiency of our model because we are still in the early stages of the tests. We hope to present interesting results in the near future.

In order to demonstrate the versatility of our proposal, other tests have been performed using another functional domain: the selection of the best players for rugby. In this application, each player's position is considered a different skill. Elementary competencies are for instance the ability to tackle and stop an opponent or to be accurate in kicking the ball. The evaluation of a player for the embodiment of a given skill is based on his performance for each criterion and on the number of selections. When the system is asked to suggest a player for a given skill, equation (3) is used. Then the propositions elaborated by each skill agent are validated (or not) by the user, the players are evaluated and the database is updated. The results are positive for the identification of players over the different iterations.

2) Actors selection simulator

Interesting results were found using the simulation tool to verify our hypothesis. The dynamic computation, with memorization, for weights of criteria is valid and convergent over time. An example of a convergence graph (here for 2 criteria over 4), across simulations, is shown in Figure 4. In this example, the weight of the first criterion is converging to 0.8, and the weight of the second one is converging to 0.6.

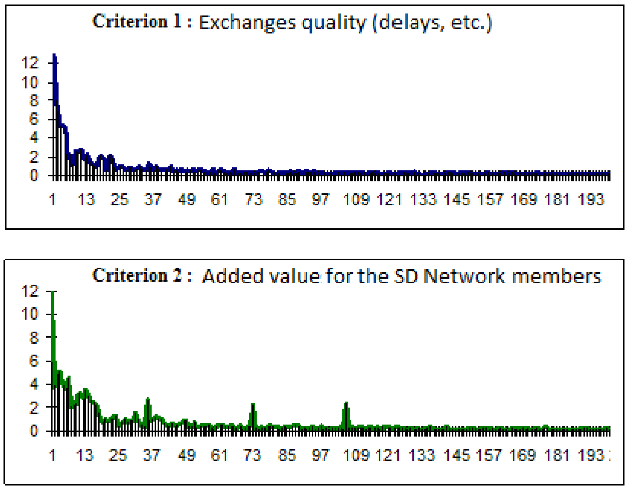


Figure 4. Convergence of criteria values

It has also been found that, for a given skill agent, the learning mechanism is efficient and can be considered as becoming "stable" when the system has stored around 120 requests and validations coming in from users. Another interesting point is the comparison of the values of criteria weight across simulations (see Figure 5).

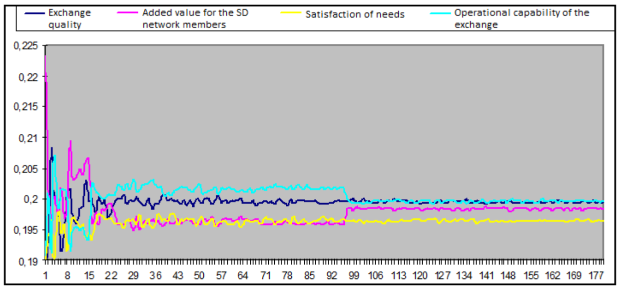


Figure 5. Compared criteria weights values

It does not seem meaningless to admit, from a theoretical point of view, that an elementary competency E_1 is "more important" than another E_2 for the global skill definition if the weight of E_1 is higher than the weight of E_2. If the graph above shows this fact, outside of a simulation process, in "real life" conditions, this observation will allow us to identify the most important elementary skills for a given skill definition. Another conclusion is that our proposal is versatile, applicable to any professional activity domain, and not only to SD projects. The example of the selection of players for rugby positions, a very different domain, is also possible as mentioned before in this article.

3) New project creations

a) Case 1: new projects from past objectives

The user request, through the *WebRequesterAgent*, is transmitted to the *Objective* agent that processes it (see Figure 6).

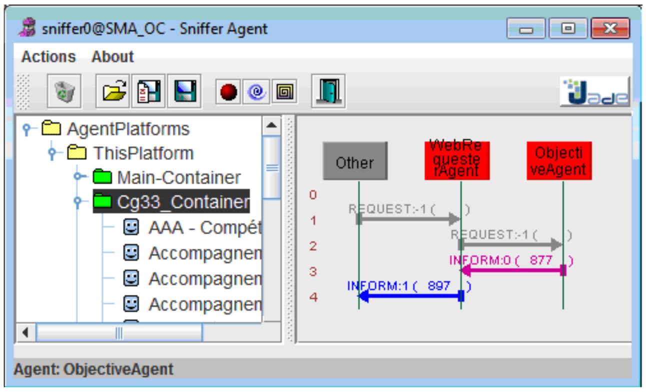


Figure 6. Case 1 - Exchanges between agents into the JADE MAS

Figure 6 is a screen copy of the JADE MAS Sniffer tool that monitors message exchanges between agents. The left part shows the MAS Agent tree where an agent is part of a container, a container part of a platform (*ThisPlatform*) and a platform part of all agent platforms (*AgentPlatForms*). The right part shows three "boxes":

- *Other*: reflects other agents within the MAS, or in this case, the JADE gateway that manages exchanges with the external users
- *WebRequesterAgent*: manages the interactions with the human user, forwarding requests to other agents, and sending back their answers.
- *ObjectiveAgent*: see next paragraph for details.

The arrows (1 to 4) show the message exchanges, with their type (REQUEST or INFORM for the answer), and their directions (from sender to receiver).

The *Objective* agent ensures the treatments, based on historical data, in retrieving project instances and related skills. At the end, it processes the answer in the form of an XML flow (see Table IV).

TABLE IV. XML FIELDS INTO THE ANSWER FLOW

XML Tag	Comment
answers	Main tag encapsulating the answers
answer	Main tag for each answer, 1 for each project
newObjective	Boolean value indicating that the project is new. Value always false here because the project is over and taken into the historical
objective	Main tag for the past project
code	The past project code within the projects database table
description	The past project textual description within the projects database table
startDate	The past project start date
endDate	The past project end date
skills	Main tag encapsulating the skills list
skill	Main tag, 1 per skill
code	The current skill code within the skills database table
description	The current skill textual description within the database table

At the front-office user level, a list of projects and skills is proposed. The new user project is then generated according to one of the two methods identified: duplicate or customize.

a) Case 2: new projects from objective domain

The user request is sent to the Objective agent that does a broadcast to all skill agents and consolidates their answers. At the end, the answer is also returned to the user as an XML flow, as already shown in case 1. The "BroadcastAgent" is a technical one and reusable by all other agents if they need to do such "broadcasting" actions in the future. See Figure 7 for a screen copy of the JADE MAS Sniffer tool. Only the right part of the window is shown for better visibility. The three first "boxes" are identical to those described for Figure 6. Please note that BroadcastAgent receives a message from an original sender, broadcasts it to all agents, and consolidates all answers into a single one. The global answer, an XML flow, is then sent back to the original sender. All other boxes on the right are all skill agents within the MAS. The screen copy proposed Figure 7 only shows two skill agents.

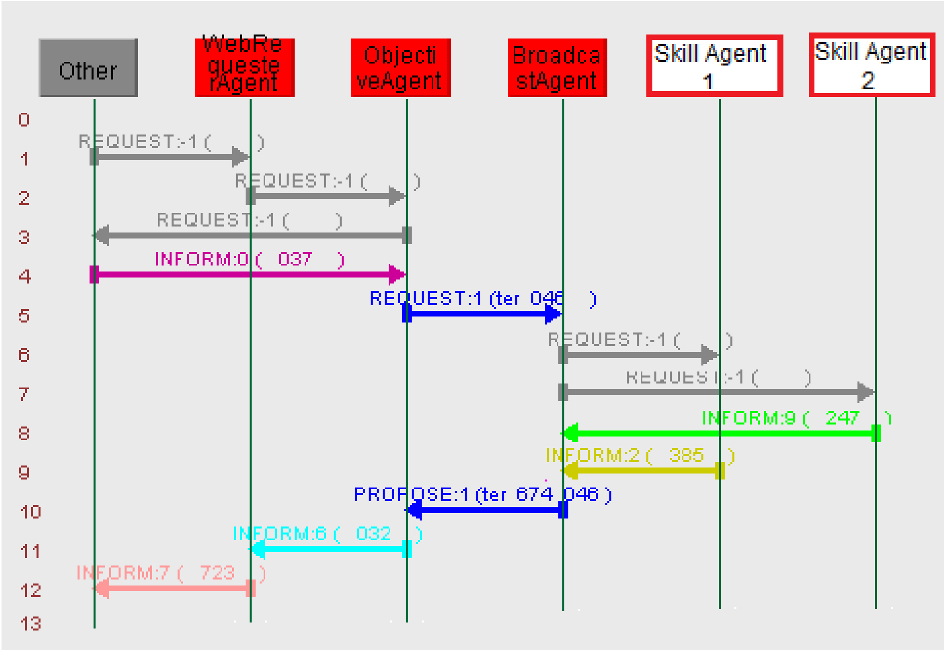


Figure 7. Case 2 - Exchanges between agents into the JADE MAS

For illustration, let us reuse the example where the objective domain is "*new means of energy production*". A project for a "*wind turbine implementation*" has five phases:

1. *Project management / implementation*
2. *Definition of power requirements*
3. *Selection of the best wind turbine technology among models and worldwide suppliers*
4. *Logistic definition (transportation organization)*
5. *Wind turbine installation*

A second project, entitled "*hydraulic micro power implementation*", shares 3 (over 5) common phases with the first project, and has three *specific ones*:

1. *Project management / implementation*
2. *Definition of power requirements*
3. *Selection of the best hydraulic micro power technology among models and worldwide suppliers*
4. *Logistic definition (transportation organization)*
5. *Hydraulic micro power installation*
6. *Technology transfer of appropriate designs to developing country manufacturers*
7. *Project formulation and appraisal for national and international aid agencies*
8. *Training on small-hydro technology and economics*

A third one is entitled: "*installation of a biodiesel generation system to power up a highway construction site*". Let us assume the project also shares the 3 common phases, and has *two specific ones*:

1. *Identify the best site*
2. *Environmental benefits evaluations*
3. *Project management / implementation*
4. *Definition of power requirements*
5. *Selection of the best technology of biodiesel generation system among models and worldwide suppliers*
6. *Logistic definition (transportation organization)*
7. *Biodiesel generation system installation*

Considering these phases as skills (of course at a high level of abstraction), let us introduce into the system a new user request where the new project is "*solar panels implementation*". This project also belongs to the objective domain "*new means of energy production*". The requested minimum value for the relevance coefficient is 75%. Thus, according to our algorithm:

- The *ObjectiveAgent* sends (broadcasts) the user request to all the skill agents within the MAS (here 3 common + 5 specifics according to the second and third project phases)
- Each skill agent computes its dedicated relevance coefficient. If this computed value is greater than the requested one, that means the skill agent "wants" to participate in the new objective, and the boolean value "*true*" is returned to the *ObjectiveAgent*
- The *ObjectiveAgent* consolidates the results where the answer is "*true*". Finally, it returns to the user the list of the skills as an XML flow:
  a - Project management / implementation
  b - Definition of power requirements
  c - Logistic definition (transportation organization)

- Through the front-office interface, the user then validates (partially or totally) the skills list to build its new project. This validation is stored in the memory.

C. Discussion

1) Skill Agents

In most SMA applications, the skills are not agents. They are typically described by behavioral rules that determine the actions of the agents [28]. The difficulty is often in making the link between tacit and explicit knowledge and learning from the real world [27, 28]. For instance, in other applications such as the management of skills in the context of e-learning, one of the main problems is to determine and make explicit the tacit knowledge that has not been understood and to adapt the courses [29]. The advantage of our approach is that it is skill centered. The skills are learning agents and their motivation is to determine the list of elementary competencies that define themselves and their relationships with the other skills. These elementary competencies usually correspond to tacit knowledge and know-hows that cannot be easily defined. One of the key ideas of our model is to consider that the weighted list of criteria defined by the users to determine the best actor for a given skill are abstractions of a hidden list of elementary competencies. The system learns from the requests of the users.

2) Skills management improvement using objectives

The proposed information system also offers an answer to the problems of each SD Network Member at CG33: they need to identify the required skills to make their project successful. Whatever the activity sector, project management is usually done through software applications, where tasks are defined and described in a static way. The definition of a new project requires the identification of human actor(s) for each of them. The first problem occurs when there is an evolution of the processes. A traditional approach is to update statically the list of tasks for the project. Settings renewals have to be done by administrators or advanced users in the project management tool itself. This update is often a generator of costs, because in some cases an external help (e.g., by the software editor) may be required. In this approach there is a lack of efficiency, inducing at least a waste of time, and sometimes some additional financial charges that could be substantial. Another problem is the management of projects dynamics according to the user needs at a given date. As a project reflects the user needs at this date, there can be as many projects as user needs expressions within the system. The global skill sharing system presented in this paper is a collaborative tool and not a static one. The new projects are built “on the fly”, from the real user needs. Thus, the new skills list is built from those available in past projects and reflects the user needs at the time of the request. As the number of projects grows in our collaborative system, the global list of skills and the objective domains evolve and may converge. At a global level, our system learns from the requests of the users and reflects the evolution of activity over time. The observation of those evolutions will drive the management of the company to put the focus on certain skills or to drop them. Our proposal therefore provides an interesting answer to the problem of skills management, using objectives, at the project and organization levels.

IV. CONCLUSION AND FUTURE WORK

A multi-agent system has been proposed for skills sharing between actors in collaborative projects in the domain of sustainable development. The key point of this work is the definition of skills as agents with their own rules for learning and evolving in an environment where actors are considered as resources for the embodiment [12] of the agents. If it is always necessary to use skills within projects, the choice of the human actor to implement them is not "suggestive and human centered" anymore. Over time, more and more objectives will be concretized through projects, and more and more information will be available to help the user. This work suggests interesting perspectives. From a professional point of view, concerning the problem of skills management, the analysis of skills applications through projects provides a wealth of information. After a period of running a project, managers and human resources management services will be able to identify the key skills, the cross-domain skills and the evolution of the “sensibility” of each skill in the professional sector processes, and all of this over time. This work may introduce real benefits into human resources management to anticipate future evolutions of needs in terms of collaborators’ profiles. Whatever the professional activity domain, the skill sharing system ensures reciprocity, human cooperation and versatility. Going further, considering that some skills may be viewed as "cross-domain" and/or "cross-companies", what about impacts on the future organization of work ?

Several issues have been identified for future works. The current tests have to be deeply validated in “real life” environment. The system has to be tested with a comprehensive list of skills (currently 110 at CG33) and elementary competencies provided by experts of the domain. A large number of evaluations are also required to test the evolution of the agents at different ages. Talking about skill management improvement, the dynamism over time is introduced by the computation of a pertinence coefficient value. At this time, this value is currently stored in the memory of each skill agent but not used in future iterations. More work has to be done to study the computation optimization of this value, and its integration in the learning mechanisms.

When talking about skill management improvement using objectives, the adaptation over time is made possible by the computation of a relevance coefficient value. This value is stored in the memory of each skill agent. One of the future direction for works is implementing complementary learning mechanisms to optimize the computation of this value over time.

REFERENCES

[1] O. Chator, J.M. Salotti, and P.A. Favier, "Multi-agent system for skills sharing in sustainable development projects," Proceedings of COGNITIVE 2013, the 5th International Conference on Advanced Cognitive Technologies and Applications, pp. 21-26, IARIA Conference, Valencia, Spain, 2013

[2] M. Bardou, "De la stratégie à l'évaluation : des clés pour réussir un Agenda 21 local," ("From the strategy to the assessment: the keys to succeed in the elaboration of a local Agenda 21") , Collection "Références" du Service de l'Économie, de l'Évaluation et de l'Intégration du Commissariat Général au Développement Durable, Paris, France, 2011

[3] J. Ferber, "Multi-Agent systems. An introduction to distributed artificial intelligence", Addison Wesley, London, UK, 1999

[4] N.R. Jennings, M. Wooldridge, and K. Sycara, "A roadmap of agent research and development," Journal of Autonomous Agents and Multi-Agent Systems, vol. 1, 1, pp. 7-38, Boston, USA, 1998

[5] G. Weiss, "Multiagent systems, a modern approach to distributed artificial intelligence," MIT Press, USA, 2013

[6] ADEME, "Bâtiment et démarche HQE," ("HQE building and methodology"), Collection Connaître pour agir, Paris, France, 2006

[7] S. Belkada, A. I. Cristea, T. Okamoto, "Measuring knowledge transfer skills by using constrained-student modeler autonomous agent," Proceedings of the IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society, 2001

[8] FIPA, "Foundation for intelligent physical agents - Abstract architecture specification (standard version)," publication of FIPA's Technical Committee, Geneva, Switzerland, 2002

[9] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE – A FIPA-compliant agent framework," 4th International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM-99, pp. 97-108, London, UK, 1999

[10] R. Pfeifer and J.C. Bongard, "How the body shapes the way we think: a new view of intelligence," The MIT Press, USA, 2007

[11] L. Shapiro, "Embodied cognition," in Oxford Handbook of Philosophy and Cognitive Science, E. Margolis, R. Samuels, and S. Stich (eds.), Oxford University Press, UK, 2010

[12] F. Varela, E. Thompson, and E. Rosch, "The embodied mind: cognitive science and human experience," MIT Press, USA, 1991

[13] M. Mulder, T. Weigel, and K. Collins, "The concept of competence in the development of vocational education and training in selected EU member states: a critical analysis," Journal of Vocational Education and Training , vol. 59, 1, pp. 67-88, 2007

[14] M. L. Putterman, "Markov decision proceses. Discrete stochastic dynamic programming," Wieley-Interscience, New York, USA, 1994

[15] M. Mitchell, "Machine learning," chap. 13 "Reinforcement learning," pp. 367-390, Ed. McGraw-Hill Higher Education, USA, 1997

[16] O. Von Neumann and O. Morgenstern, "Theory of games and economic behaviour," Princeton University Press, USA, 1944

[17] L. Shapley, "Stochastic games," Proceedings of National Academy of Science, 39, pp. 1095–1100, USA, 1953

[18] C. Boutilier, T. Dean, and S. Hanks, "Decision theoretic planning: structural assumptions and computational leverage," Journal of Artificial Intelligence Research, USA, 1999

[19] E. Plaza and S. Ontañon, "Cooperative multi-agent learning in Adaptive Agents and Multi-Agent Systems," Lecture Notes in Computer Science, vol. 2636, pp. 1-17, Spain, 2003

[20] D. McSherry, "Completeness criteria for retrieval in recommender systems," Advances in Case-Based Reasoning, 8th European Conference (ECCBR'06), pp. 9-29, Turkey, 2006

[21] H. Mühlenbein, "How Genetic Algorithms Really Work: 1. Mutation And Hill Climbing," Manner, R. and Manderick, B. (eds), Proceedings of the Second Conference on Parallel Problem Solving from Nature, Elsevier Science, vol. 2, pp. 15-25, Belgium, 1992

[22] S.C. Shiu, K. Shiu, D.S. Yeung, C.H. Sun, and X. Z. Wang, "Transferring case knowledge to adaptation knowledge : An approach for case-base maintenance," Computational Intelligence", vol. 17, issue 2, P 295-314., 2001

[23] J.R. Quinlan, "C4.5 : Programs for Machine Learning," Proceedings of European Conference on Machine Learning, pp. 3-20, Austria, 1993

[24] P. Domingos, "Unifying instance-based and rule-based induction," Machine Learning, vol. 24, issue 2, pp. 141-168, 1996

[25] M.J. Zaki, C.J. Hsiao, "CHARM : An efficient algorithm for closed itemset mining," Proceedings of the Second SIAM International Conference on Data Mining, pp. 12-28, USA, 2000

[26] G. Rocher, "The definitive guide to Grails," Apress, New York, USA, 2006

[27] S. Goderis, "On the separation of user interface concerns: a programmer's perspective on the modularisation of user interface code," Ph.D. thesis, VrijeUniversiteit Brussels, Belgium, 2007

[28] H. Friedrich, O. Rogalla and R. Dillmann, "Integrating skills into multi-agent systems," Journal of Intelligent Manufacturing, vol. 9, 2, pp. 119-127, Karlsruhe, Germany, 1998

[29] A. Garro, L. Palopoli, "An XML multi-agent system for e-Learning and skill management,", proceeding of: Agent Technologies, Infrastructures, Tools, and Applications for E-Services, NODe 2002 Agent-Related Workshops, Erfurt, Germany, 2002

New Autonomous Decentralized Structure Formation Based on Huygens' Principle and Renormalization for Mobile Ad Hoc Networks

Kenji Takagi
Graduate School of System Design,
Tokyo Metropolitan University
Tokyo, Japan
Email: kenji.takagi.tmu@gmail.com

Chisa Takano
Graduate School of Information Sciences,
Hiroshima City University
Hiroshima, Japan
Email: takano@hiroshima-cu.ac.jp

Masaki Aida
Graduate School of System Design,
Tokyo Metropolitan University
Tokyo, Japan
Email: aida@tmu.ac.jp

Makoto Naruse
National Institute of Information
and Communications Technology (NICT)
Tokyo, Japan
Email: naruse@nict.go.jp

***Abstract*—This paper proposes an autonomous distributed algorithm that can construct spatial structures for clustering in mobile ad hoc networks. Since the topology of a mobile ad hoc network changes frequently, a fast, light-weight, and autonomous clustering mechanism is required. However, existing autonomous clustering mechanisms are based on differential equations and thus demand a lot of calculations for generating the spatial structures that yield clustering. This paper proposes an autonomous clustering algorithm that is based on Huygens' principle and renormalization. The most remarkable characteristics of our proposed scheme are light calculation loads and fast convergence on the cluster structures. We verify the basic characteristics of the proposed scheme. In addition, we introduce an algorithm to control the number of generated clusters in the framework of the proposed scheme by introducing a logarithmic representation of network state.**

Keywords*—*Mobile ad hoc network, Autonomous decentralized control, Clustering, Huygens' principle, Renormalization.

I. Introduction

This paper is an extended version of the paper presented at COLLA2013 [1]. The significant progress of this paper from the earlier version is that we can control the number of cluster even if we cannot know the network state and its metric in advance. Details are explained in the body of this paper.

In large-scale communication networks, hierarchical architectures are effective for scalable network control. Let us consider how hierarchical structure can be introduced into networks. For the case of fixed networks, we can set the desired hierarchical structure at the time of designing the networks. However, this is not possible for mobile networks since their topology changes dynamically. A typical example the mobile ad hoc network (MANET) [2]. A MANET consists of mobile terminals that work as routers. That is, each terminal offers routing functions and data forwarding functions. Two terminals can directly communicate if their coverage areas of wireless communications overlap. If the areas do not overlap, the terminals communicate but relaying data through the terminals between them, they can establish multihop communications. To achieve multihop communications, routing is one of most important issues in MANETs.

The most primitive route finding approach is called flooding [3]. In flooding, the sender terminal sends route finding packets to all adjacent terminals, which resend them to all their adjacent terminals until at least one copy of the packet reaches the destination terminal. The total amount of route finding packets sent in MANETs increases exponentially with network size (the number of terminals). It is known as the broadcast storm problem [4]. So one of the challenges in MANETs, realizing scalable routing control [5][6], is best addressed by setting a hierarchical structure through clustering [7][8].

Hereafter, we call a MANET terminal a node. An autonomous clustering mechanism for generating a hierarchical structure must offer several characteristics, as follows:

- Each node acts autonomously based on local information about its neighboring nodes.
- The generated cluster structure should reflect the state information of the network (e.g., battery power of nodes).
- The generated cluster structure should be flexible so that it can adapt to the dynamic environment.
- The convergence time of clustering should be sufficiently shorter than the timescale of topology change caused by node movement. This is because clustering should dynamically adapt to the network topology.
- Action rules of each node should be as simple as possible in order to reduce the battery power consumed by computation or processing at the node.

Since MANETs are expected to be an effective communication tool after serious disasters, the above requirements are essential for realizing clustering in ad hoc networks.

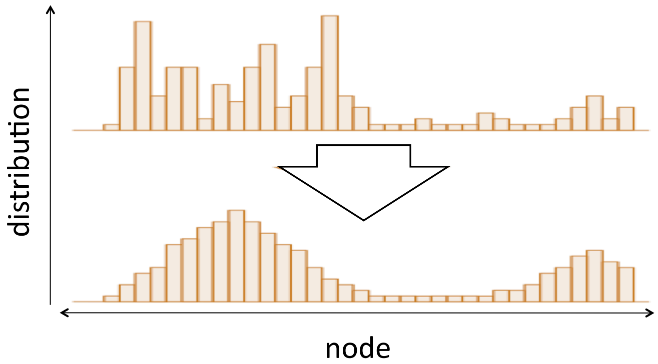


Figure 1. Concept of cluster forming.

Neglia et al. proposed a clustering mechanism based on reaction-diffusion equations [9]. Let us call it the bio-inspired approach. This is an application of Turing patterns and satisfies the first requirement listed above. Takano et al. has proposed a clustering mechanism based on the Fokker-Planck equation and includes the drift motion given by back-diffusion [10][11][12]. Let us call it the back-diffusion based approach. The back-diffusion based approach satisfies the first two requirements listed above. In addition, Hamamoto et al. recently propose a mechanism that guarantees the asymptotic stability of the cluster structure generated by the back-diffusion based approach; they showed that their mechanism can satisfy the first three requirements listed above [13].

The guarantee mechanism of asymptotic stability, in particular, implies the possibility of new autonomous clustering mechanisms. Specifically, we might be able to make a new clustering algorithm that satisfies all the requirements listed above by replacing the back-diffusion algorithm with a simple and fast-converging rule. This is because the guarantee mechanism of asymptotic stability does not depend on details of the clustering mechanism. In this paper, we use Huygens' principle as the simple and fast-converging rule, and propose a new clustering mechanism that satisfies all the requirements listed above.

The paper is organized as follows: Section II explains the concept of cluster formation and the guarantee mechanism of asymptotic stability, which is the foundation of this research. Section III proposes an autonomous clustering mechanism based on Huygens' principle. Section IV shows cluster structures generated by our proposed scheme using numerical examples and verifies that they reflect the network condition. Section V introduces an algorithm to control the number of generated clusters in the framework of the proposed scheme by introducing a logarithmic representation of network state. In addition, we show a control method of the number of clusters. The conclusion is discussed in Section VI.

II. Preliminary

This section shows the framework of autonomous decentralized clustering and the related mechanism to stabilize the cluster structure.

A. Concept of Cluster Formation

In our clustering model, each node has a certain value and cluster formation is conducted by the distribution of the values of nodes. The initial value is determined by considering a certain network metric (e.g., battery power of each node). The clustering algorithm extracts a coarse grained spatial structure from the initial distribution of the values and this procedure corresponds to clustering. Figure 1 shows an example of cluster formation in a simple 1-dimensional network. The horizontal axis represents node position, and the vertical axis represents the value of the distribution for each node. The upper-half of Figure 1 represents the initial distribution, which reflects a certain network state. The lower-half of Figure 1 represents the generated coarse grained spatial structure. Each peak of the coarse grained distribution corresponds to the center of a cluster, and cluster structures reflect the initial condition.

The back-diffusion based approach is an example of this mechanism, and has a relatively faster convergence rate than the bio-inspired approach [14]. However, this clustering mechanism does not consider change from the initial condition, and so cannot adapt to the dynamic environment common to MANETs. That is why this mechanism does not satisfy the third requirement listed in the previous section.

B. Guarantee mechanism of Asymptotic Stability

To adapt the spatial structure to dynamic environments, the guarantee mechanism of the asymptotic stability of cluster structures was proposed by Hamamoto et al. [13]. In this mechanism, cluster structure generation can adapt to changes in network state; the mechanism generates stable spatial structures if the network state is fixed. As an alternative approach, Takayama et al. proposed the self-adjustment approach to stabilize the cluster structure [15]. However, it has a restriction that is applicable only to the back-diffusion based clustering. Thus, we focus on [13] in this paper.

Let us consider a one-dimensional network model for simplicity, and let $q(i,t)$ be the distribution value at node position i at time t. The distribution value, $q(i,t)$, determines the cluster structure. As an example, the initial condition $q(i,0)$ and cluster structure $q(i,t)$ obtained at time t are shown in Figure 1. The conventional back-diffusion based approach described in Takano et al. [12] presents a rule that governs the temporal evolution of the distribution, $q(i,t)$. However, as shown in the previous section, it is difficult to guarantee the stability of $q(i,t)$ for large t. In other words, the cluster structure is not stable in a dynamic environment.

Let us consider discrete time t_k $(k = 1, 2, \dots)$, and distribution $q(i,t_k)$. Since we need an autonomous decentralized algorithm, the temporal evolution of distribution $q(i,t_k)$ is determined by its local information. By introducing the temporal evaluation operator of $\mathcal{T}$, the temporal evolution is formally described as

$$q(i,t_{k+1}) = \mathcal{T}(q(i-1,t_k),\, q(i,t_k),\, q(i+1,t_k)). \quad (1)$$

This rule states that the distribution of node i at the next time is completely determined by the values of the present distribution at node i and its adjacent nodes.

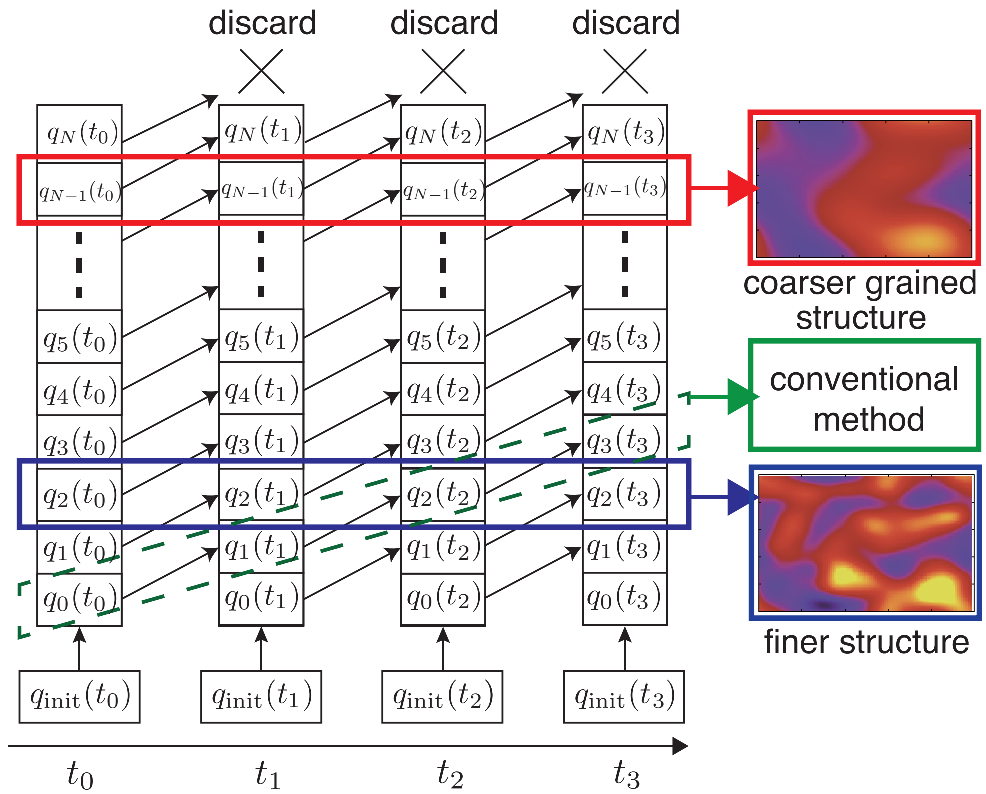


Figure 2. Outline of the guarantee mechanism of asymptotic stability.

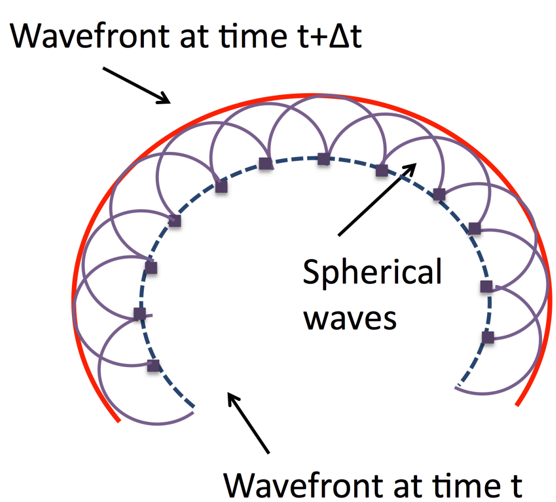


Figure 3. Example of wave propagation obeying Huygens' principle.

To guarantee asymptotic stability, we consider a vector of the distribution. Each node i has the following $N+1$ dimensional vector

$$\mathbf{q}(i,t_k) = \{q_0(i,t_k),\, q_1(i,t_k),\, \ldots,\, q_N(i,t_k)\}. \tag{2}$$

Here, we define the rule for the temporal evolution of the vector $\mathbf{q}(i,t_k)$. Let $q_{\mathrm{init}}(i,t_k)$ be the distribution describing the network state at time t_k. Then we set

$$q_0(i,t_{k+1}) = q_{\mathrm{init}}(i,t_{k+1}). \tag{3}$$

If $q_{\mathrm{init}}(i,t_k)$ is independent of time, $q_0(i,t_{k+1}) = q(i,0)$, that is, the initial condition of the conventional mechanism. Note that, in general, we allow the time-dependence of $q_{\mathrm{init}}(i,t_k)$. Next, for $q_{n+1}(i,t_{k+1})$ $(n = 0,\, 1,\, \ldots,\, N-1)$, we set

$$q_{n+1}(i,t_{k+1}) = \mathcal{T}(q_n(i-1,t_k),\, q_n(i,t_k),\, q_n(i+1,t_k)). \tag{4}$$

Although the above rule may look complicated, we can easily understand it through graphical representation. Figure 2 explains the temporal evolution of vector (2) at node i. The horizontal axis represents discrete time as $t_0,\, t_1,\, \ldots,$ and $q_{\mathrm{init}}(t_k)$ is a certain value expressing the network state of a node at time t_k. Each component of the vector is a value of the distribution and (4) is the temporal evolution rule for the nth component $(n = 1,\, 2,\, \ldots,\, N-1)$. The temporal evolution of each component will be updated to the upper-right component in Figure 2. The component at the bottom, $q_0(i,t_{k+1})$, is overwritten by the network condition $q_{\mathrm{init}}(i,t_{k+1})$ at the present time. The component at the top will be discarded.

The temporal evolution of the conventional mechanism corresponds to the sequence indicated by the green broken line in Figure 2. In the guarantee mechanism of asymptotic stability, we focus on the sequence of the same vector components. If we choose small n for the nth component, we obtain a finer-grained spatial structure as indicated by the blue line. A large n yields a coarse grained spatial structure as indicated by the red line.

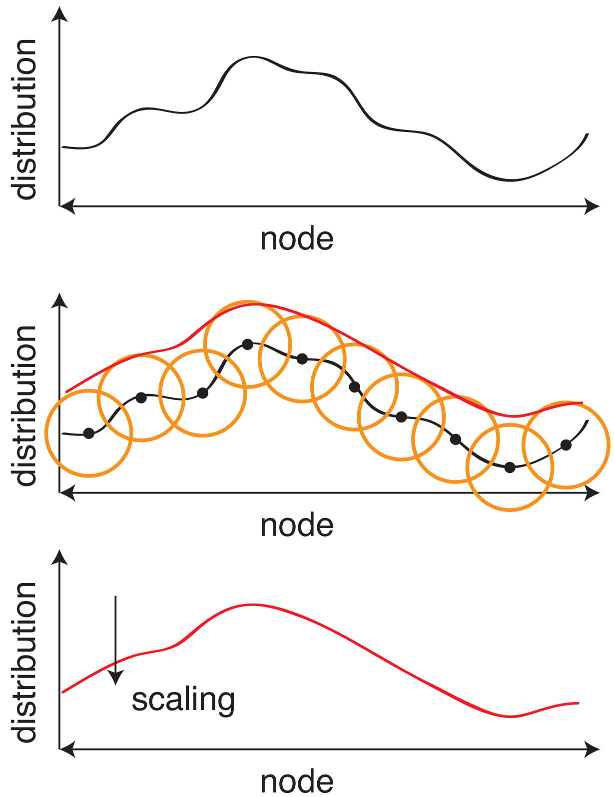


Figure 4. Renormalization transformation as per Huygens' principle.

III. Design of the Autonomous Structure Formation Technology based on Huygens' Principle

In this section, we first propose a new autonomous decentralized clustering based on Huygens' principle and its renormalization transformation. Second, we investigate the attenuation of the cluster structure caused by the fixed point of the renormalization transformation, and introduce a procedure of amplification for avoiding the attenuation.

A. Huygens' principle and Renormalization

Huygens' principle, or the Huygens-Fresnel principle describes the temporal evolution of the wavefront and can explain the laws of reflection and refraction. Figure 3 shows an example of wave propagation obeying Huygens' principle. Let us consider a wavefront at present time t. In Huygens' principle, spherical waves originate from all points on the wavefront and the envelope of these spherical waves gives the temporal evolution of the wavefront at time $t + \Delta t$.

Renormalization is a way to extract simple and important macroscopic characteristics from a large-scale and complex system, and its procedure is called renormalization transformation. This procedure is suitable for generating a simple cluster structure extracted from the spatial structure of the network state. Renormalization transformation is defined as the combination of coarse-grained transformation and scaling. In this paper, we adopt the renormalization transformation based on Huygens' principle as temporal evolution operation $\mathcal{T}$. Specifically, each node generates spherical wave-like information for temporal evolution of the distribution. Concrete procedures of the renormalization transformation are shown below.

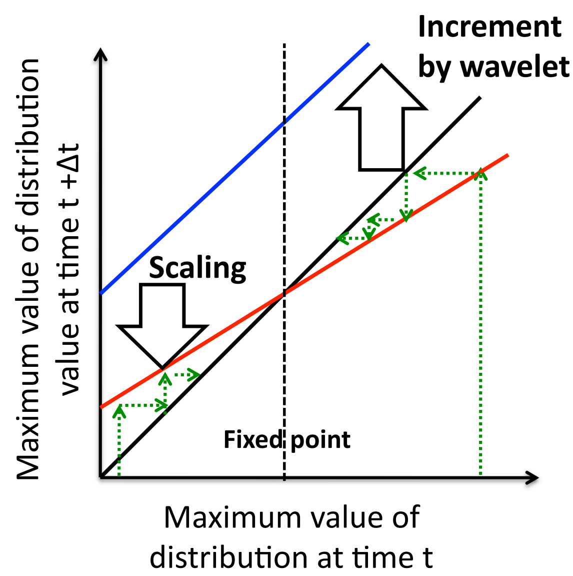


Figure 5. Fixed point under renormalization transformation.

Let us consider a one-dimensional network and a distribution on the network. The panel at the top of Figure 4 shows an example of the distribution at the present time. We consider the shape of the distribution as the wavefront. The panel at the middle of Figure 4 shows the temporal evolution of the wavefront as given by Huygens' principle. This procedure has a smoothing effect in which the fine-grained structure in the shape of the distribution becomes smooth. The temporal evolution of the distribution causes an increase in the value of the distribution, that is, the wavefront proceeds upward. In order to compensate for this increase, we introduce scaling as shown in the panel at the bottom of Figure 4. We define the renormalization transformation as the combination of such temporal evolution and scaling.

Let the distribution value of node i at time t_k be $q(i, t_k)$, and let the set of nodes that are adjacent to node i at time t_k be $M(i, t_k)$. In addition, $\tilde{q}(i, j, t_{k+1})$ is the wavefront of the spherical wave at node i at time t_{k+1} that originated from node j at time t_k. Our renormalization transformation is expressed as

$$q(i, t_{k+1}) = \frac{1}{b} \max_{j \in M(i,t_k)} \tilde{q}(i, j, t_{k+1}), \tag{5}$$

where, the maximizing operation in (5) means Huygens' principle; it determines the most advanced wavefront of spherical waves that originated from the node itself and its neighborhood, b is the scaling parameter.

Next, let us consider a method of tuning parameter b using Figure 5. In Figure 5, the horizontal axis represents the maximum value of the distribution at time t , and the vertical axis represents the maximum value of the distribution after renormalization transformation at time $t + \Delta t$. Here, the black

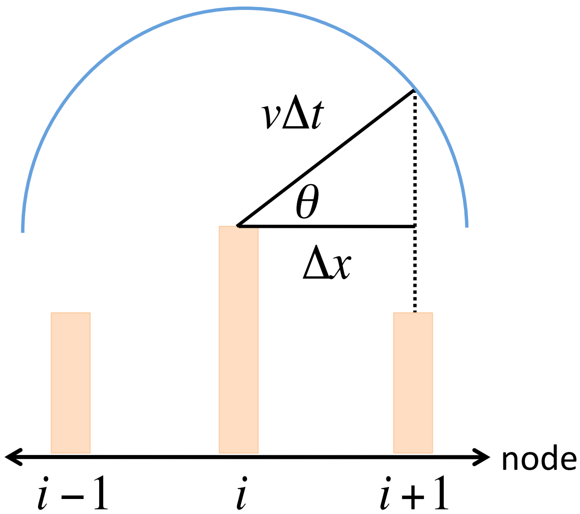


Figure 6. The wavefront of the spherical wave reaches an adjacent node.

line shows the linear equation $y = x$. This linear equation corresponds to the top of Figure 4. Next, the maximum value of the distribution is increased obeying Huygens' principle. This increment can be expressed by $v\Delta t$ and the blue line, which is shifted up from $y = x$ by $v\Delta t$. The red line is obtained from the blue line by dividing the latter by the scaling parameter b ($b > 1$). There is an intersection point between the red line and the black line, and this intersection point is given by

$$p^* = \frac{v\,\Delta t}{b-1}. \tag{6}$$

The value of interaction point p^* is the fixed point under renormalization transformation. We can understand that the maximum value of distribution $q(i, t_k)$ converges to p^* with iterations of the temporal evolution, regardless of initial condition $q(i, 0)$. Hence, the method of tuning parameter b does not need to be highly accurate.

Next, we consider the concrete form of $\tilde{q}(i, j, t_{k+1})$. Let the propagation speed of the spherical wave be v, the distance between two adjacent nodes be Δx, and the interval of the temporal evolution (renormalization transformation (5)) be Δt (i.e., $t_{k+1} - t_k = \Delta t$). Here, Δx is not physical distance but is a kind of hop count, so we choose $\Delta x = 1$. We consider the situation wherein the temporal evolution (5) is determined only by adjacent nodes, v is chosen as $1 \le v\,\Delta t < 2$. As shown in Figure 6, the wave front of the spherical wave originating from node i influences both node i and its adjacent nodes. They are expressed as

$$\tilde{q}(i \pm 1, i, t_{k+1}) = q(i, t_k) + v\,\Delta t\,\sin\theta, \tag{7}$$
$$\tilde{q}(i, i, t_{k+1}) = q(i, t_k) + v\,\Delta t, \tag{8}$$

where θ is a constant and, from $v\,\Delta t\,\cos\theta = \Delta x$,

$$\theta = \arccos\left(\frac{\Delta x}{v\,\Delta t}\right). \tag{9}$$

Since v, Δt, and $\sin\theta$ are constants and we can know them in advance, temporal evolution (5) is a simple operation.

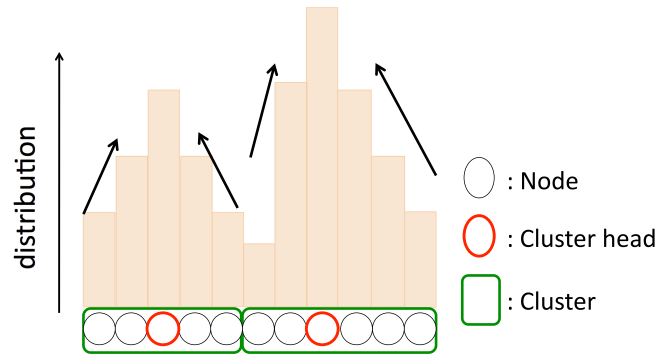


Figure 7. Determination of the cluster and cluster head.

B. Amplification of the Range of the Distribution

Our renormalization transformation (5) makes the distribution flat and we can obtain a coarse-grained spatial structure. However, different from physical phenomena, there are situations that the distribution does not change when the difference in distribution values is small. This is because the positions of nodes in the network are discrete. If the distribution value at a node can affect that of the adjacent node, the following relation is required;

$$|q(i \pm 1, t_k) - q(i, t_k)| > v\,\Delta t\,(1 - \sin\theta). \tag{10}$$

When smoothing proceeds and the condition (10) is no longer met, the two adjacent nodes no longer interact and the distribution is unchanged. To avoid this phenomenon, we introduce amplification of the range of the distribution in addition to renormalization transformation (5). The additional operation is

$$q(i, t_{k+1}) \leftarrow p^* + a\,\left(q(i, t_{k+1}) - p^*\right), \tag{11}$$

after renormalization transformation (5). This operation means that the difference between the distribution value and p^* is amplified by a factor of a. Here, the aforementioned $p^* = v\,\Delta t/(b-1)$ is the fixed point of renormalization transformation, and also is the convergence point. Note that, the value of parameter a should be chosen as $a > b$.

Finally, we explain how to determine clusters and cluster heads from the generated spatial structure (Figure 7). By following the direction of the steepest gradient of the distribution, we can find a node with local maximum value. We define it as the cluster head, and the nodes belonging to the same cluster head belong to the same cluster.

IV. Evaluation for State Dependent Characteristics of Clusters

This section shows cluster structures generated by our proposed scheme using numerical examples and verifies that they reflect the network condition. Our simulation programs are written by C language.

A. Simulation Model and Parameters

In this subsection, we explain a simulation model and parameters used in our verification. In this verification, we use a unit disk graph having a torus boundary as the network model.

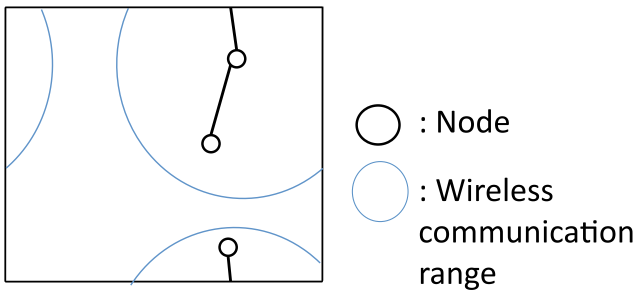


Figure 8. Unit disk graph having torus boundary.

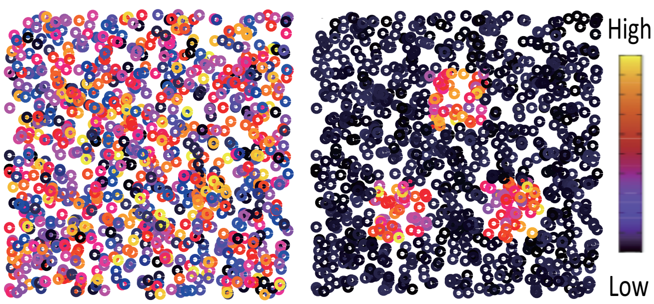


Figure 9. Initial condition.

We can configure the unit disk graph through the following procedure. We set the position of each node randomly, and set a circle of a certain radius around each node. If the circle of a node encloses another node(s), we set a link between the center node of the circle and each node lying within the circle. Since we can assume that circle radius is wireless communication range, the unit disk graph is a model that can express MANET topology. Figure 8 shows an example of the torus boundary by focusing on the wireless communication range of a certain node. The reason of setting the torus boundary condition is to eliminate the effect of the network edge, and to concentrate our attention on the characteristics of clustering mechanism itself. In this evaluation, we set $1,000$ nodes on a plane of 1 km $\times$ 1 km, and use 60 m as the wireless communication range.

In order to verify the state dependent characteristics of the cluster structures generated, we use two initial distributions as shown in Figure 9 where distribution values are described by a color map. The left figure in Figure 9 expresses a randomized state. The initial values of $q_{\rm init}(i, 0)$ for all node positions, i, are random values that obey a uniform distribution with range $[0, 10]$. The right figure in Figure 9 expresses a spatially structured state. Three areas have relatively high values, and the values of these areas are determined by random values that obey a uniform distribution with range $[5, 10]$. The values of other area are determined by random values that obey a uniform distribution with range $[1, 2]$. If, for example, the value represents battery energy, the randomized state does not express any power-supply information whereas the spatially structured state indicates three power-supply zones. The parameters used in the evaluation are shown in Table I.

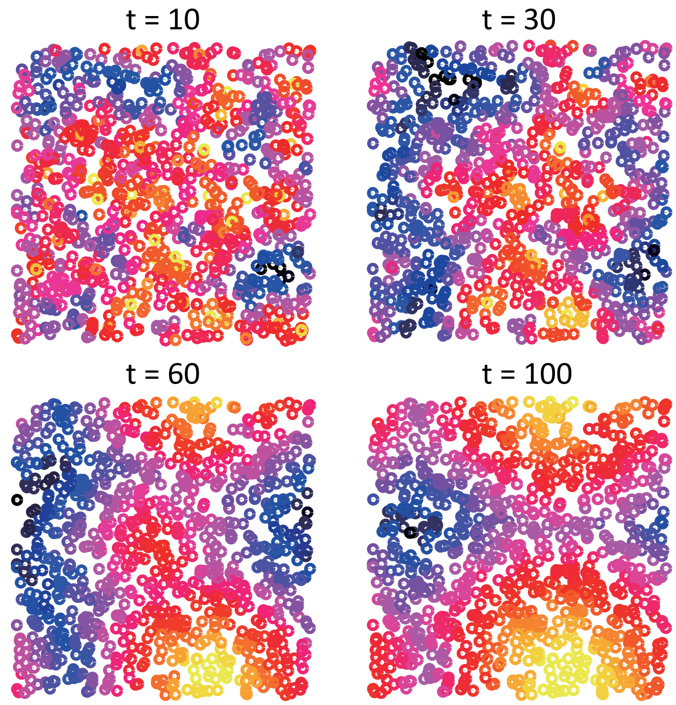


Figure 10. The cluster structures generated from a randomized state.

B. Evaluation

Figures 10 and 11 show the cluster structures generated from a randomized state and a spatially structured state, respectively. The four panels of each figure shows the number of iterations or, equivalently, the position of the vector component in the guarantee mechanism of the asymptotic stability of cluster structures. From these results, if we choose few iterations, we get a finer-grained cluster structure, and if we choose more iterations, we get a coarse grained cluster structure. We can also recognize that the cluster structures reflect the spatially configured state of the initial condition. Since the initial condition reflects the network state (e.g., battery power of each node), it means the cluster structure generated by our proposed scheme can reflect the state information of the network.

V. Controllability of The Number of Clusters Based on the Guarantee mechanism of Asymptotic Stability

This section investigates the clusters generated by our proposed scheme combined with the guarantee mechanism of asymptotic stability and reveals a technological problem

TABLE I. Parameter setting.

parameter	value
v	1.5
b	1.1
a	1.2
Δx	1.0
Δt	1.0

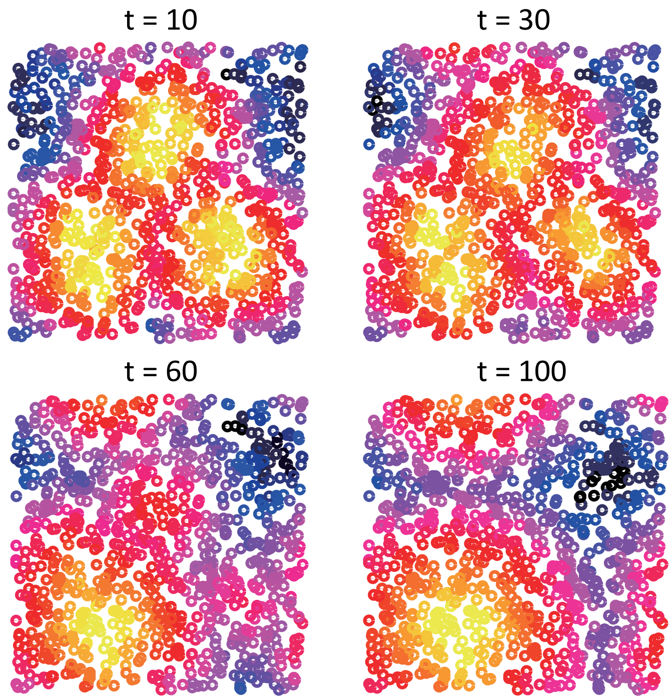


Figure 11. The cluster structures generated from a spatially structured state.

for ensuring control of the number of clusters. To solve this problem, we propose a control method of the number of clusters by introducing a logarithmic representation of network state.

A. Metric for Describing Network State

Let us consider what metrics can be used to describe network state. For example, we can use node battery power as the initial condition of nodes that reflects the state of the network. There are many ways to express the battery power in numerical values: ampere-hour [Ah], milli-ampere hour [mAh], coulomb [C], etc. Incidentally, 1 Ah = 1,000 mAh = 3,600 C. Thus, the ranges of the initial distribution may quite different depending on the metric used, even if the target networks are in the same environment. In addition, it is possible that some other network state metrics might be included as a part of the distribution value. Therefore, it is difficult to specify the metric for describing network state in advance.

In our proposed clustering scheme combined with the guarantee mechanism of asymptotic stability, the number of clusters are controlled by choosing the number of iterations (i.e., the position of the vector component of (2)). In order to realize the control of the number of clusters based on our framework, for any given iteration number, the number of generated clusters should be independent of the metric.

B. Evaluation for Dependence Characteristics on the Range of the Initial Distribution

In this subsection, we investigate the convergence of the number of generated clusters with respect to the range of the initial distributions. We use the same network model and parameter setting as used in the previous section. The values of the initial distribution $q_{\text{init}}(i, 0)$ for all node positions, i, are given by a random values that obey a uniform distribution; four kinds of uniform distributions are examined. Their ranges are $[1,\, 5]$, $[10,\, 5 \times 10]$, $[10^2,\, 5 \times 10^2]$ and $[10^3,\, 5 \times 10^3]$. The difference in ranges means the difference in the metrics. An example of an initial condition is shown in the left pane of Figure 9.

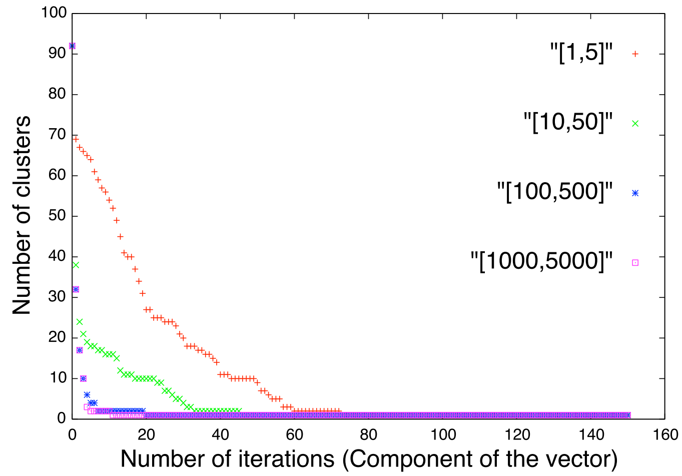


Figure 12. The number of generated clusters for randomized initial conditions w.r.t. the number of iterations.

Figure 12 shows the temporal evolutions of the number of clusters obtained from four different initial conditions. From these results, we can recognize that the number of clusters strongly depends on the range of the initial distribution. Also, the relationship between the number of clusters and the iteration times depends on the range of the initial distribution. Since we cannot know the network state and its metric in advance, we cannot control the number of clusters.

The cause of the above problem is the excessive sensitivity of cluster formation to the range of the distribution. The mechanism of the excessive sensitivity with respect to the range of the distribution can be recognized through Figure 13. Figure 13 shows the behaviors of cluster formation obtained from our proposed scheme for the initial distributions with large and small ranges. First, each node performs temporal evolution obeying Huygens' principle, and then scaling. Small-valued nodes, which are next large-valued nodes, are greatly influenced by the latter, and the difference in values between them is strongly decreased. Therefore, if the range of the initial distribution is large, the distribution is rapidly harmonized.

C. Proposal of Control Method of the Number of Clusters

In order to avoid the problem caused by the difference in the ranges of distribution, we redefine the initial condition. The details are as follows. We do not use the network condition directly as the initial condition, but instead use

$$q(i, 0) = \log(q_{\text{init}}(i, 0)). \tag{12}$$

In the vector formulation, we replace (3) with

$$q_0(i, t) = \log(q_{\text{init}}(i, t)). \tag{13}$$

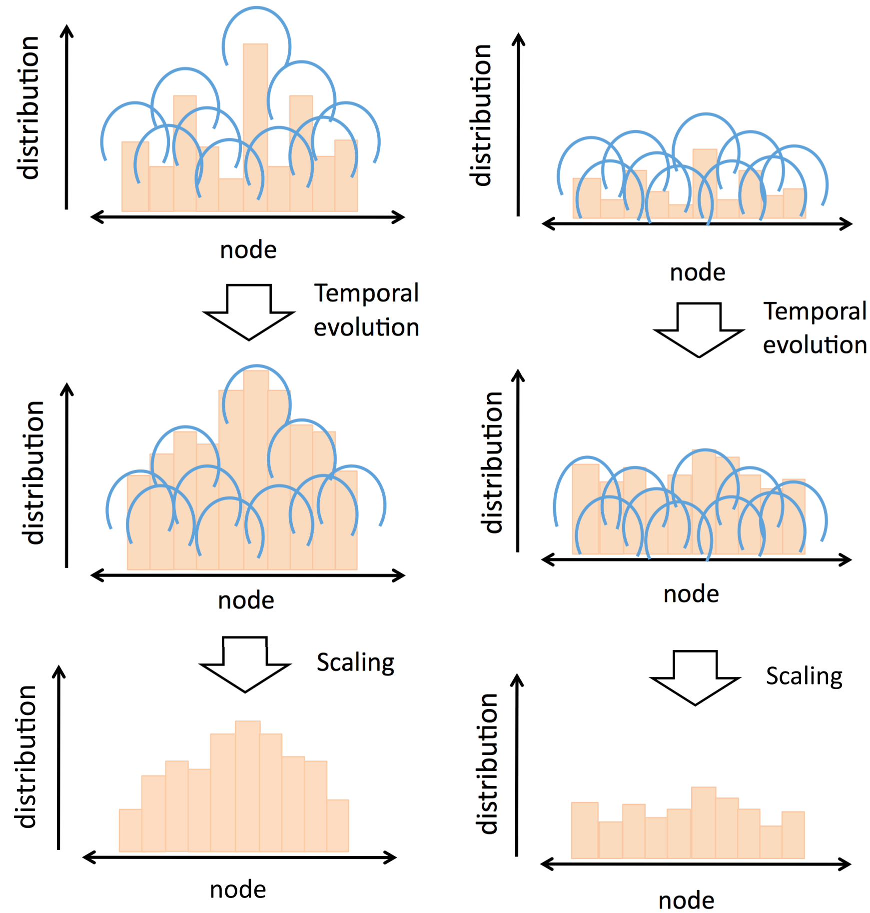


Figure 13. An example of the behavior of the proposed mechanism on two different distribution ranges.

The reasons for introducing a logarithmic function are as follows:

- It is possible to maintain the magnitude relation of the values of the original initial distribution.
- If the value of the original initial distribution is large, its new equivalent value is smaller in the sense of the ratio.

Let us consider the situation wherein the range of initial condition is given by $[Ax, Bx]$. Where x is a positive constant representing the difference between metric values. Here, we define the distribution range function $R[q]$ as the difference between the maximum and minimum values of distribution q. The range of the conventional distribution, q_{init}, is expressed as follows:

$$R[q_{\mathrm{init}}] = (B - A)\,x. \tag{14}$$

This means that the range of the distribution depends on x. On the other hand, the range of the new distribution $\log q_{\mathrm{init}}$ is expressed as follows:

$$R[\log(q_{\mathrm{init}})] = \log(Bx) - \log(Ax) = \log\left(\frac{B}{A}\right). \tag{15}$$

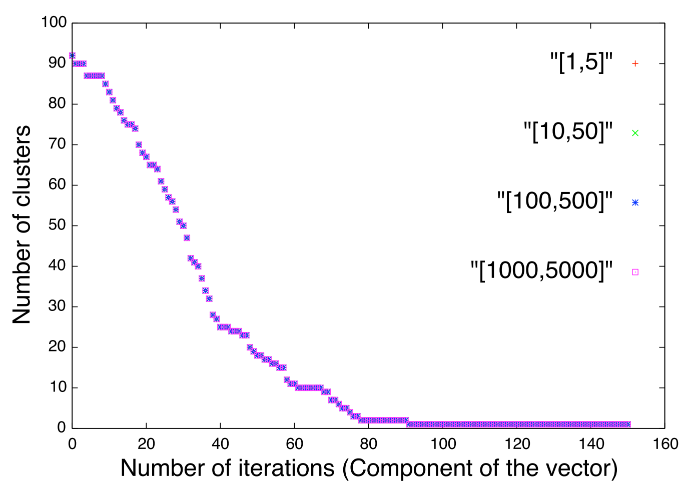


Figure 14. The number of clusters generated using logarithmic function from randomized initial conditions, w.r.t. the number of iterations.

This means that the range of the distribution is independent of x. It also means that the range of the redefined initial distribution is independent of the metric used.

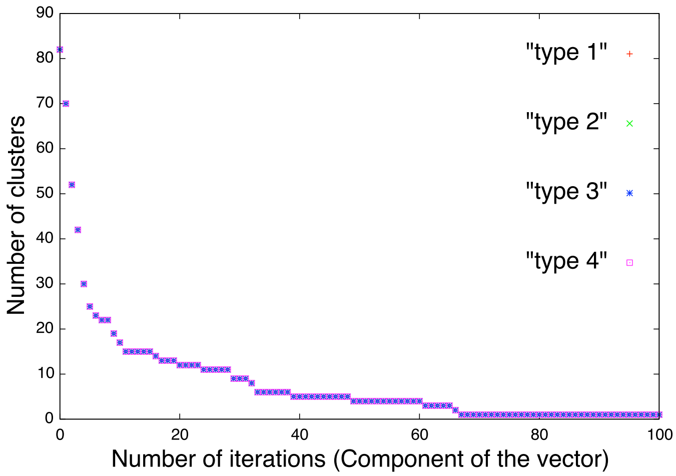


Figure 15. The number of clusters generated using logarithmic function from the initial condition having patterned structure, w.r.t. the number of iterations.

D. Evaluation for Controllability of the Number of Clusters

Figure 14 shows similar evaluations to Figure 12 by using the redefined initial condition (13). We recognize that the impact of the initial condition is sufficiently weakened. In particular, all initial conditions yield completely the same result. This means that we have a robust clustering mechanism that can control the number of clusters by appropriately choosing the number of iterations or the component of the vector.

Next, we evaluate an effectiveness of our control method on other type of initial distribution. We use the initial distribution having spatially structured state (right figure in Figure 9). Also, we prepare the range of initial conditions in the same manners as the previous evaluation. Specifically, their ranges are as follows. The values on the three areas are determined by random values that obey a uniform distribution with range $[5, 10]$. The values on the other areas are also determined by random values that obey a uniform distribution with range $[1, 2]$. Let us call this initial condition type 1. Next, we define the initial condition type 2, whose ranges are calculated by raising the ranges of type 1 by a factor of ten. In the same manner, we define type 3 and type 4.

Figure 15 shows the temporal evolution in the number of clusters obtained from our scheme with the logarithmic initial conditions for the four different ranges. From these results, we can recognize that the number of clusters is independent of the range of initial distribution even if the initial distribution is spatially structured. Therefore, our proposed scheme can be expected to control the number of clusters even if the initial distribution has a complex spatial structure.

VI. Conclusion and Future Work

In this paper, we proposed an autonomous clustering mechanism based on Huygens' principle for MANETs. For verification, we used a unit disk graph to evaluate the characteristics of the proposed scheme. The benefits of the proposed algorithm lie in its simplicity and its ability to form the spatial structure reflecting the initial condition of network states. To control the number of clusters generated, the number should be independent of the metric representing the initial network condition. However, unfortunately, the convergence speed of cluster configuration depends strongly on the range of the initial distribution that describes the network state. Since we cannot know the distribution value for each node in advance, the difference in convergence speed makes it impossible to control the number of clusters. To avoid this problem, we introduced new distribution defined by the logarithm of the original distribution. Consequently, the difference in convergence speed is significantly weakened, and the number of clusters becomes controllable. The above characteristics are suitable for clustering in MANET. As future work, we will consider the response of our mechanism to dynamic environments.

Acknowledgment

This work was supported by a Grant-in-Aid for Scientific Research (B) No. 23300031 (2011-2013) from the Japan Society for the Promotion of Science.

References

[1] Kenji Takagi, Masaki Aida, Chisa Takano, and Makoto Naruse, "A proposal of new autonomous decentralized structure formation based on Huygens' principle and renormalization," in Proceedings of the 3rd International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA 2013), Jul. 21–26, 2013, Nice, France, pp. 75–80, ISBN: 978-1-61208-287-5, ISSN: 2308-4227.

[2] Charles E. Perkins, Ad Hoc Networking, Addison-Wesley, Jan. 2001, ISBN: 978-0201309768.

[3] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das, "Ad hoc on-demand distance vector (AODV) routing," Request for Comments 3561, Jul. 2003.

[4] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu, "The broadcast storm problem in a mobile ad hoc network," Wireless Networks, Kluwer Academic Publishers, vol. 8, issue 2–3, Mar. 2002, pp. 153–167, ISSN: 1572-8196.

[5] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in Proceedings of MOBICOM'98 of the $4th$ annual ACM/IEEE international conference on Mobile computing and networking, Oct. 25–30, 1998, Dallas, TX, USA, pp. 151–162, ISBN: 1-58113-142-9.

[6] Stefaan Seys and Bart Preneel, "ARM: Anonymous routing protocol for mobile ad hoc networks," International Journal of Wireless and Mobile Computing, Inderscience Publishers, vol. 3, issue 3, Oct. 2009, pp. 145–155, ISSN: 1741-1084.

[7] Stefano Basagni, "Distributed clustering for ad hoc networks," in Proceedings of the 4th International Symposium on Parallel Architectures, Algorithms, and Networks, 1999 (I-SPAN'99), Jun. 23–25, 1999, Perth/Fremantle, WA, USA, pp. 310–315, ISSN: 1087-4089.

[8] Tomoyuki Ohta, Shinji Inoue, Yoshiaki Kakuda, and Kenji Isida, "An adaptive multihop clustering scheme for ad hoc networks with high mobility," IEICE Transactions on Fundamentals, vol. E86-A, no. 7, Jul. 2003, pp. 1689–1697, ISSN: 0916-8508.

[9] Giovanni Neglia and Giuseppe Reina, "Evaluating activator-inhibitor mechanisms for sensors coordination," in Proceedings of the 2nd IEEE/ACM International Conference on Bio-Inspired Models of Network, Information and Computing Systems (Bionetics 2007), Dec. 10–12, 2007, Budapest, Hungary, pp. 129–133, ISBN: 978-963-9799-05-9.

[10] Masaki Aida, "Using a renormalization group to create ideal hierarchical network architecture with time scale dependency," IEICE Transactions on Communications, vol. 95-B, no. 5, May 2012, pp. 1488–1500, ISSN: 0916-8516.

[11] Chisa Takano, Masaki Aida, Masayuki Murata, and Makoto Imase, "New framework of back diffusion-based autonomous decentralized control and its application to clustering scheme," in Proceedings of the IEEE Globecom 2010 Workshop on the Network of the Future

(FutureNet III), Dec. 6–19, 2010, Miami, FL, USA, pp. 362–367, ISBN: 978-1-4244-8863-6.

[12] Chisa Takano, Masaki Aida, Masayuki Murata, and Makoto Imase, "Proposal for autonomous decentralized structure formation based on local interaction and back-diffusion potential," IEICE Transactions on Communications, vol. E95-B, no. 5, May 2012, pp. 1529–1538, ISSN: 0916-8516.

[13] Ryo Hamamoto, Chisa Takano, Kenji Ishida, and Masaki Aida, "Guaranteeing asymptotic stability of clustering by autonomous decentralized structure formation," in Proceedings of the 9th IEEE International Conference on Autonomic and Trusted Computing (ATC 2012), Sep. 4–7, 2012, Fukuoka, Japan, pp. 408–414, ISBN: 978-1-4673-3084-8.

[14] Kenji Takagi, Yusuke Sakumoto, Chisa Takano, and Masaki Aida, "On convergence rate of autonomous decentralized structure formation technology for clustering in ad hoc networks," in Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ADSN 2012), Jun. 18–21, 2012, Macau, P.R.China, pp. 356–361, ISBN: 978-1-4673-1423-7, ISSN: 1545-0678.

[15] Hiroki Takayama, Sota Hatakeyama, and Masaki Aida, "Self-adjustment mechanism guaranteeing asymptotic stability of clusters formed by autonomous decentralized mechanism," Journal of Communications, vol. 9, no. 2, Feb. 2014, pp. 180–187, ISSN: 1796-2021.

Applications of Languages with Self-Interpreters to Partial Terms and Functional Programming

Lev Naiman
Department of Computer Science
University of Toronto
Toronto, Canada
Email: naiman@cs.toronto.edu

Abstract **—Those programming languages that contain self-interpreters have the added power of reflection, and allow dynamically controlling execution. In a logical language a complete self-interpreter is necessarily inconsistent. However, we demonstrate a logical language with a reasonably complete self-interpreter. We argue for its use as a simple formalism for reasoning about partial terms, and functional languages that allow both general recursion and dependent types. Since refinements of programming specifications often include partial terms, they need to be handled using formal rules. Likewise, we show formal rules for handling general recursion consistently in a simple language. Moreover, we demonstrate how to use an interpreter to reason about lazy evaluation. We argue that the interpreter can be integrated within theorem provers.**

Keywords ***— logic; partial-terms; theorem prover; two-valued logic; expression interpreter; functional programming; general recursion; lazy evaluation***

I. Introduction

In this paper we argue that logics for programming must be able to cope with general recursion and partial terms, and that an interpreter [1] is a viable solution.

General recursive and partially recursive functions naturally occur in programs. This is an effect of recursively defined datatypes such as trees, and the computation paths that arise in sufficiently complex programs. Aside from necessary complexity while programming, partially recursive functions are often the result of computation that is non-terminating. While often this indicates programmer error is is undesirable, there are many cases where a non-terminating program is intentional. This includes any application code that waits and responds to user input, and some semi-decision procedures. Therefore, general recursion is not a property that should be excluded, but rather desirable in a functional language for the ease of use of programmers.

Despite their power and expressiveness, general and partially recursive functions pose challenges for a number of theories of programming and practical tools. The issues present themselves both in the difficulty of proof of their properties and with the possible partiality or at worst inconsistency that they introduce. For example, the language Gallina within the interactive theorem prover Coq [2] requires that all functions terminate. Non-terminating programs would introduce logical inconsistency. Likewise, theories of programming such as Morgan's Programming from Specifications [3] exclude non-terminating programs from its standard theory and require a proof of termination for all programs. In the Vienna Development Method [4] non-terminating computation corresponds to partial functions. In a well-typed or dependently typed language this partiality naturally arises when a function is required to produce a certain result for a pre-condition stronger than **true**.

The occurrence of partial terms are not limited to non-terminating functions. There are often cases where expressions within programs do not denote a value. For example the indexing of a sequence with a negative number results in an error in most programming languages. Formal reasoning about partial expressions often occurs when using a formal programming theory, even one that requires termination. Both issues of partial terms and general recursion pose a similar type of problems to a theory: they produce expressions to which no formal rules apply, and make proofs of intuitively simple theorems impossible. At worst, the result is inconsistency.

It is crucial for a formal program theory to consistently and elegantly deal with general recursion and partial terms, introducing the minimal amount of extra values and extra theory to do so. The interpreter formalism can be applied for reasoning about both of these features, and is an extension of the interpreter presented in [1].

A. General Recursion

For formal typed functional languages recursion is often restricted. This is because logically reasoning about functions with both constructive types and general recursion is inconsistent. For example, from the following definition:

$$f : nat \to nat \tag{1}$$

$$f = \lambda x : nat \cdot 1 + f\, x \tag{2}$$

It is immediately clear that $f\, n = f\, n + 1$, and since $f\, n : nat$ we have the contradiction that $1 = 0$. There are a number of methods to maintain consistency that either restrict recursion, require functions to be constructive, or require proof that both argument and result of a function is a value [5]. A constructive type theory is desirable in order to perform effective error checking statically. Dependent types are more expressive, and allow type information to encode invariants.

B. Partial Terms

In programming specifications and their refinements we commonly encounter partial terms. Partial terms are defined as expressions that fail to denote a value. A term t in a theory T is partial if there are no laws in T that apply to t. An example is where a function or an operator is applied to an argument outside of its domain, such as $1/0$. We also say that a formula e is unclassified in theory T if it is neither classified as a theorem or an anti-theorem. Such expressions are present in proofs of programs due to the partial functions and operators that are often used in specifications. Borrowing an example from [6], we might implement the difference function as follows (where the domain of *diff* is integers, and the assumed theory is arithmetic and first-order two-valued logic).

$$\mathit{diff}\; i\; j = \mathbf{if}\; i = j\; \mathbf{then}\; 0\; \mathbf{else}\; (\mathit{diff}\; i\; (j+1)) + 1\; \mathbf{fi} \tag{3}$$

We would like to prove

$$\forall i, j : \mathit{int} \cdot i \geq j \Rightarrow (\mathit{diff}\; i\; j) = i - j \tag{4}$$

but when trying to simplify this expression instantiated with 1 and 2 respectively for i and j we get

$$\begin{aligned} & 1 \geq 2 \quad \Rightarrow \quad (\mathit{diff}\; 1\, 2) = 1 - 2 \\ = \;& \mathbf{F} \quad\;\;\; \Rightarrow \quad (\mathit{diff}\; 1\, 2) = -1 \end{aligned} \tag{5}$$

and we cannot apply any laws at this point to simplify it further. A law would allow simplifying the expression to true, but it requires that both operands be boolean. The expression *diff* $1\,2$ is a partial term because no laws apply to it. For this reason we cannot use any law to conclude that $(\mathit{diff}\; 1\, 2) = -1$ is a boolean, even though it has the form $X = Y$. Tools that reason with such expressions must be based on formal rules in order to have confidence in their proofs. We propose a character-string interpreter to solve this problem.

The rest of the paper is organized as follows: in Section II we examine the existing approaches in the literature to cope with partial terms. In Section III we describe the background theories we use to define the interpreter in Section IV. Section V shows how the interpreter can be used to cope with partial terms. Section VI describes other benefits of the interpreter when constructing theories. Section VIII describes how we can extend the definition of the interpreter to be more expressive.

II. Current Approaches to Partial Terms

One approach to resolve partial terms is to make all terms denote. Formally this means that for each partial term such as $x/0$, a law must exist saying which set of values that expression is a member of. This set of values is assumed to already be defined in the logic, as opposed to newly created values. In this case there could be a law defined saying that $\forall x : \mathit{int} \cdot x/0 : \mathit{int}$. This is the approach used in the programming theory of [3]. Such laws do not explicitly say what value a partial term is equal to, and this can cause certain peculiar and possibly unwanted results such as $0/0 = 0$ being a theorem.

TABLE I
Three-Valued Boolean Operators

	T	**F**	$\perp$
$\neg$	**F**	**T**	$\perp$

	TT	**TF**	**FT**	**FF**	**T**$\perp$	$\perp$**T**	$\perp$**F**	**F**$\perp$	$\perp\perp$
$\vee$	**T**	**T**	**T**	**F**	**T**	**T**	$\perp$	$\perp$	$\perp$
$\wedge$	**T**	**F**	**F**	**F**	$\perp$	$\perp$	**F**	**F**	$\perp$

$$\begin{aligned} & 0 \\ =\; & 0 \times (1/0) \\ =\; & 1 \times (0/0) \\ =\; & 0/0 \end{aligned} \tag{6}$$

This approach can be slightly modified and the value of partial terms can be fixed. However, this might cause some unwanted properties. In the case of division by zero a choice of 42 as used in [7] cannot be allowed due to inconsistency.

In [8], the authors point out that underspecification alone may cause problems. If we allow domains of single elements then these problems can go as far as inconsistency. The semantic model of our interpreter uses underspecification, but not exclusively. In some cases, similarly to LPF, the interpreter would leave some expressions unclassified. One way of finding a model for partial functions in set theory is the standard approach of mapping any unmapped element from the domain to a special value, usually called $\perp$ [9]. The denotational semantics for a generic law for equality are extended with this value, and in this particular model $7/0 = 5/0$ would be a theorem (assuming strict equality). However, a user of a logic that includes the interpreter would not need to perform any calculations that concern this extra value.

The Logic of Partial Terms (LPT) [10], [11] is an example of a logic that does not include the undefined constant. It does however include a definedness operator $\downarrow$. In this theory the specialization law $(\forall x \cdot A(x)) \Rightarrow A(v)$ requires that v be defined. The basic logic of partial terms (BPT) [12] is a modification of LPT, and relaxes the previous requirement for some laws. It allows for reasoning with non-terminating functional programs. Some logics such as [13] include multiple notions of equality to be used in calculations. This may complicate the laws of quantifiers.

Another approach to deal with partial terms is a non-classical logic such as LPF [7] with more than two values. In these logics the truth table of boolean operators is usually extended as in Table I (where $\perp$ represents an "undefined" value, and the column heads are both of the arguments to the operator). In this logic the expression $0/0 = 1$ would not be classified to one of the boolean values, but would rather be classified as $\perp$. Undefinedness is either resolved by the boolean operators or is carried up the tree of the expression. Some three valued logics have a distinct undefined value for each value domain, such as integers and booleans.

Three and more valued logics have varied useful applications. However, a drawback of using a logic with multiple truth values is that certain useful boolean laws no longer hold. This is particularly true of the law of the excluded middle, $\forall x : bool \cdot x \vee \neg x$, which in a three value logic can be modified to $\forall x : bool \cdot x \vee \neg x \vee undefined(x)$. In the Logic of Computable Functions (LCF) [14] there is a $\bot_t$ value for each type t, requiring the modification of several laws. Another issue of multiple valued logics is that not knowing the value of an expression seems to be pushed one level up; attempting to formalize these extra values will result in a semantic gap. There are always expressions that must remain unclassified for a theory to remain consistent.

A further method of dealing with partial terms is conditional, or short-circuit operators [15]. This approach is similar to those logics with three values, since it gives special treatment to partial terms. Boolean operators have an analogous syntax $a \,\mathbf{cor}\, b$, $a \,\mathbf{cand}\, b$, $a \,\mathbf{cimp}\, b$, etc. In these expressions if the first value is undefined, then the whole expression is undefined. These conditional operators are not commutative.

For many of these non-classical logics the authors of [16] demonstrate a relationship, and how to transform undefined terms in one logic to another in a similar method to data-refinement.

III. Background Theories

We introduce two theories from [17] that we will use to define the interpreter.

A. Bunch Theory

A bunch is a collection of objects. It is different from a set, which is a collection of objects in a package. A bunch is instead just those objects, and a bunch of a single element is just the element itself. A number, character or boolean is an element. Every expression is a bunch, but not all bunches are elementary. Here are some bunch operators.

$A\,,B$	A union B	(7)
$A\,‘B$	A intersect B	(8)
$A : B$	A in B, or A included in B	(9)
$¢A$	cadinality of A	(10)

If x is an element, then $¢x = 1$. The empty bunch, whose cardinality equals zero, is the constant $null$. The union of two elements x, y is not an element iff $x‘y = null$. Both bunch union and intersection are symmetric, associative, and idempotent. The definition of bunches essentially gives algebraic properties to a comma as an operator. Operators such as a comma, colon, and equality apply to whole bunches, but some operators apply to their elements instead. In other words, they distribute over bunch union. For example

$$\begin{aligned} & 1 + (4, 7) \\ =\ & 1 + 4\ ,\ 1 + 7 \\ =\ & 5\,, 8 \end{aligned} \tag{11}$$

Bunch comprehension is denoted with the section sign §. For element x, bunches A and B, and predicate f, § is defined as follows:

$$(\S v : null \cdot f\,v) \quad = \quad null \tag{12}$$
$$(\S v : x \cdot f\,v) \quad = \quad \mathbf{if}\; f\,x\; \mathbf{then}\; x\; \mathbf{else}\; null \tag{13}$$
$$(\S v : A, B \cdot f\,v) \quad = \quad (\S v : A \cdot f\,v), (\S v : B \cdot f\,v) \tag{14}$$

Where nat is the bunch of naturals, we define the notation $x, ..y$, read as “x to y” as

$$x, ..y \ = \ \S i : nat \cdot x \le i < y \tag{15}$$

Bunch distribution is similar to a cross-product in set theory. Sets do not distribute over bunch union, and set brackets can be placed around a bunch to form a set (which itself is an element). For example, $\{null\}$ is the empty set, and $¢\{null\} = 1$. Set comprehension $\{x : D | f\,x\}$ is an abbreviation for $\{\S x : D \cdot f\,x\}$.

B. String Theory

A string is an indexed collection of objects. It is different from a list or ordered pair, which are indexed collections of objects in a package. A string of a single item is just that item. The simplest string is the empty string, called nil. Strings are joined together, or concatenated with the semicolon operator to form larger strings. This operator is associative but not commutative. The string $0; 1$ has zero as the first item and one as the second. For a natural number n and a string S, $n{*}S$ means n copies of S. Let nat be the bunch of natural numbers. The copies operator is defined as follows.

$$0{*}S = nil \tag{16}$$
$$\forall\, n : nat \cdot (n+1){*}S = (n{*}S); S \tag{17}$$

Strings can be indexed, and their length can be obtained with the length operator (↔).

S_n	S at index n	(18)
$\leftrightarrow S$	length of S	(19)

A semicolon distributes over bunch union, and so does an asterisk in the left operand. Similarly to bunches, a number, character, or boolean is an item. If x is an item, then $\leftrightarrow x = 1$, and only the string nil has length zero. The concatenation of two items is not an item. Note that $null$ is not an item, and that $\leftrightarrow null = null$. Operators and functions, whose domains include only items, distribute over concatenation. For example

$$(0; 2; 4) + 1 \ = \ 1; 3; 5 \tag{20}$$

Since a string S can be thought of as a function that maps natural numbers from $0,.. \leftrightarrow S$ to the items of S, quantifiers can be lifted to apply to strings.

$$\Sigma S \;=\; \Sigma n : 0,.. \leftrightarrow S \cdot S_n \tag{21}$$
$$\forall S \;=\; \forall n : 0,.. \leftrightarrow S \cdot S_n \tag{22}$$

Analogously to the bunch notation $x,..y$ the notation $x;..y$ for items x, y, z is defined as

$$x;..x = nil \tag{23}$$
$$x;..(x+1) = x \tag{24}$$
$$(x;..y)\ ;\ (y;..z) = x;..z \tag{25}$$

Similarly to the relationship between sets and bunches, strings can be pakcaged into lists. Lists are denoted with square brackets, and operators of lists are

$$[S] \qquad \text{List containing S} \tag{26}$$
$$[S] + [T] \;=\; [S;T] \qquad \text{List Concatenation} \tag{27}$$
$$[S]n \;=\; [S_n] \qquad \text{List Indexing} \tag{28}$$

Some examples of the operators defined are

$$\begin{aligned} \leftrightarrow (7;1;0) &= 3 \\ (7;1;0)_0 &= 7 \\ 1;(5,17);0 &= (1;5;0),(1;17;0) \\ 3{*}(0;1) &= 0;1;0;1;0;1 \\ (0,1){*}(0;1) &= 0{*}(0;1), 1{*}(0;1) = nil, 0;1 \end{aligned} \tag{29}$$

The prefix "copies" operator $*S$ is defined to mean $nat{*}S$, or informally the bunch of any number of copies of S. Finally, we introduce characters, which we write with double-quote marks such as "a", "b", etc. To include the open and close double-quote characters we escape them with a backslash: "\“". Strings that contain exclusively character strings are sometimes abbreviated with a single pair of quotes: "abc" is short for "a"; "b"; "c". Let the bunch of all characters is called $char$. Then the bunch of all two-character strings is $char;char$.

Bunch and string theory are used because they allow for compact language definitions. For example, denoting the collection of naturals greater than zero in set theory can be done by writing $\{n : nat | n > 0\}$. In bunch theory it can be written as $nat + 1$. We can of course define an addition operator that distributes over the contents of a set, but the benefit of bunch theory (and analogously string theory) is that no such duplication is necessary. This built-in distributivity comes at a cost: the cost is that at times it is required to prove that some bunches are elements. Consider the bunch $bool$ defined as $bool = \mathbf{T}, \mathbf{F}$. Then we prove

$$\neg bool \tag{30}$$
$$= \neg(\mathbf{T}, \mathbf{F}) \tag{31}$$
$$= \mathbf{F}, \mathbf{T} \tag{32}$$
$$= bool \tag{33}$$

However, this does not mean that bunch theory is inconsistent. In order to simplify $bool = \neg bool$ to $\mathbf{F}$, we must know that $bool$ is an element.

IV. Defining the Interpreter

An interpreter is very similar to a semantic valuation function, except that it does not require a universe of values. In addition, it will be extended to interpret a language that includes the interpreter symbol itself, which if naïvely done causes inconsistency. While standard notation does not explicitly distinguish logic from meta-logic (unless different operators are used), we put quotes around the interpreted language. Since the interpreter essentially encodes meta-logic, we would like to keep it simple in the sense that it should introduce as few new operators as possible. It should also preserve the properties of existing operators. In this way we both avoid a separate meta-language, and do all reasoning within a single logic. In the literature authors often use one set of symbols for the meta-logic operators and another for the object logic. We use character strings instead both for clarity, and in the case where we wish to use the logic to study itself. Where as multiple level of meta-logic might require multiple sets of symbols, reasoning with the interpreter only requires adding more quotes. We take the idea of the character-string predicate of Hehner [18], and we extend it to be a general interpreter for any expression in our language. To maintain consistency we exclude the interpreter itself from the interpreted language in this section. The interpreter, which we call $\mathcal{I}$ is an operator which applies to character strings and produces an expression. The interpreter can be thought of as unquoting a string. We first define our language as a bunch of character strings.

Let $char$ be the bunch of all character symbols, let $alpha$ be the bunch of character symbols in the English alphabet, and let nat be the bunch of naturals. We have the following definitions

$$\begin{aligned} digit \quad &= \quad \text{“0”}, \text{“1”}, \text{“2”}, \text{“3”}, \text{“4”}, \\ &\quad\;\; \text{“5”}, \text{“6”}, \text{“7”}, \text{“8”}, \text{“9”} && (34) \\ var \quad &= \quad alpha; {*}alpha && (35) \\ num \quad &= \quad digit; {*}digit && (36) \\ uniops \quad &= \quad \text{“}\neg\text{”}, \text{“}-\text{”}, \text{“}\forall\text{”}, \text{“}\exists\text{”}, \text{“}\Sigma\text{”} && (37) \\ binops \quad &= \quad \text{“}=\text{”}, \text{“}\wedge\text{”}, \text{“}\vee\text{”}, \text{“}\Rightarrow\text{”}, \text{“}\Leftarrow\text{”}, && (38) \\ &\quad\;\; \text{“};\text{”}, \text{“}-\text{”}, \text{“}+\text{”}, \text{“ ”}, \text{“}\in\text{”} \\ charstring \quad &= \quad \text{“\textbackslash“”}; {*}char; \text{“\textbackslash””} && (39) \end{aligned}$$

We define our language $lang$ to be the following bunch of strings.

$$\begin{aligned}
&var, num, string, \text{“}\mathbf{T}\text{”}, \text{“}\mathbf{F}\text{”} : lang \\
&\text{“}\langle\text{”}; var; \text{“ : ”}; lang; \text{“}\rightarrow\text{”}; lang; \text{“}\rangle\text{”} : lang \\
&\text{“(”}; lang; \text{“)”} : lang \\
&\text{“\{”}; lang; \text{“\}”} : lang \\
&\text{“(”}; uniops; lang; \text{“)”} : lang \\
&\text{“(”}; lang; binops; lang; \text{“)”} : lang
\end{aligned} \tag{40}$$

Here we have defined a language that includes boolean algebra, numbers, logical quantifiers, functions, and strings. This language is fully bracketed for simpler laws, and non-bracketed expressions should be read as abbreviations. The language is defined similarly to how a grammar for a language would be given. Function syntax is $\langle v : D \rightarrow B\rangle$, where the angle brackets denote the scope of the function, and v is the introduced variable of type D. We treat quantifiers as operators that apply to functions. The quantifiers $\exists$ and $\forall$ give boolean results. When we use more standard notation such as $\forall v : domain \cdot body$ we mean it as an abbreviation for $\forall\langle v : D \rightarrow B\rangle$.

The interpreter is intuitively similar to a program interpreter: it turns passive data into active code. Our interpreter turns a text (character string) that represents an expression into the expression itself. The interpreter is defined very closely to how $lang$ was defined. The laws are as follows.

$$\begin{aligned}
&\mathcal{I}\,\text{“}\mathbf{T}\text{”} = \mathbf{T} \\
&\mathcal{I}\,\text{“}\mathbf{F}\text{”} = \mathbf{F} \\
&\forall\, s : num \cdot \forall\, d : digit \cdot \mathcal{I}(s; d) = (\mathcal{I}\,s) \times 10 + (\mathcal{I}\,d) \\
&\forall\, s, t : lang \cdot \mathcal{I}\,(\text{“}\langle a : \text{”}; s; \text{“}\rightarrow\text{”}; t; \text{“}\rangle\text{”}) = \langle a : \mathcal{I}\,s \rightarrow \mathcal{I}\,t\rangle \\
&\forall\, s : lang \cdot\ \mathcal{I}\,\text{“(”}; s; \text{“)”} = \mathcal{I}\,s \quad \wedge \\
&\qquad \mathcal{I}\,\text{“\{”}; s; \text{“\}”} = \{\mathcal{I}\,s\} \quad \wedge \\
&\qquad \mathcal{I}\,(\text{“}\neg\text{”}; s) = \neg(\mathcal{I}\,s) \quad \wedge \\
&\qquad \mathcal{I}\,(\text{“}-\text{”}; s) = -(\mathcal{I}\,s) \quad \wedge \\
&\qquad \mathcal{I}\,(\text{“}\forall\text{”}; s) = \forall(\mathcal{I}\,s) \quad \wedge \\
&\qquad \mathcal{I}\,(\text{“}\exists\text{”}; s) = \exists(\mathcal{I}\,s) \quad \wedge \\
&\qquad \mathcal{I}\,(\text{“}\Sigma\text{”}; s) = \Sigma(\mathcal{I}\,s) \\
&\forall s, t : lang \cdot \mathcal{I}\,(s; \text{“}=\text{”}; t) = (\mathcal{I}\,s) \wedge (\mathcal{I}t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}\wedge\text{”}; t) = (\mathcal{I}\,s) = (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}\vee\text{”}; t) = (\mathcal{I}\,s) = (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}\Rightarrow\text{”}; t) = (\mathcal{I}\,s) \Rightarrow (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}\Leftarrow\text{”}; t) = (\mathcal{I}\,s) \Leftarrow (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“;”}; t) = (\mathcal{I}\,s); (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}-\text{”}; t) = (\mathcal{I}\,s) - (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}+\text{”}; t) = (\mathcal{I}\,s) + (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“}\in\text{”}; t) = (\mathcal{I}\,s) \in (\mathcal{I}\,t) \quad \wedge \\
&\qquad \mathcal{I}\,(s; \text{“ ”}; t) = (\mathcal{I}\,s)\ \ (\mathcal{I}\,t) \\
&\forall\, s : {*char} \cdot \mathcal{I}\,(\text{“}\backslash\text{“”}; s; \text{“}\backslash\text{””}) = s
\end{aligned} \tag{41}$$

To save space we leave out the interpretation of each digit. For scopes the introduced variable must be an identifier, and the expression $\mathcal{I}a$ in that position would not satisfy this requirement. We instead have a law for only the identifier a, and other identifiers can be obtained through an application of a renaming law.

Note that we defined $lang$ as a bunch of texts, and not the expressions themselves. When these texts are interpreted, the results are expressions or values in the language. The text “2” is in $lang$, but not the value 2. The interpreter is similar to a function of strings and distributes over bunch union. It is possible to have a logical language to parallel the texts in $lang$; all the expressions in the language which do not contain $\mathcal{I}$ can then be denoted as $\mathcal{I}\,lang$. In this paper we leave out some operators from $lang$, such as the ones in bunch theory.

Note that unlike a semantic valuation function the interpreter does not necessarily map every string in the language to a value. Rather, we later introduce generic laws that reason with these partial terms directly. Lastly, we will show how the interpreter can be included in the interpreted language without inconsistency.

A. Variables

One significant change that we allow in our logic is for variables. We say that a variable with the name a is an abbreviation for $\mathcal{I}$“a”, and similarly for all other variable names. Although in our initial definition we excluded the interpreter from the interpreted strings, we later show in Section VIII how we can extend our language to safely include the interpreter.

There is an important consequence of making variable syntax more expressive: function application and variable instantiation is no longer a decidable procedure in general. This is because deciding whether two variable strings are equal is now as difficult as all of proving. However, this does not pose a problem for the implementation of function application along with the interpreter in a theorem prover. The simple solution is that whenever we see an interpreter in the body, we do not apply the function; it is treated as a syntactic variable that can only be replaced by its interpretation. We argue that this rarely hinders the use of the interpreter, since in the sub-language that does not include the interpreter users can do all calculations exactly as before. In the case where reasoning with the interpreter is desired, standard proof obligations can be generated and discharged.

We finish this section by noting that we could have simplified the definition considerably if we had a prefix language. All operator interpretation could be compressed to a single law, and some bracket characters removed.

V. Resolving Partial Terms With the Interpreter

Our solution to reasoning with partial terms is neither at the term or propositional level. We rather say that some operators, such as equality or bunch inclusion are generic. For example, here are two of the generic laws for equality.

$$\forall a, b : lang \cdot \mathcal{I}\,(a; \text{“}=\text{”}; b) : bool \qquad \text{Boolean Equality} \tag{42}$$

$$\forall a : \mathcal{I}\,lang \cdot a = a \qquad \text{Reflexivity} \tag{43}$$

The first law says that any equality is a boolean expression, similarly to the Excluded Fourth Law in LPF which implies an equality is either true, false or undefined [6]. The arguments can be any expressions in the interpreted language. For a simple formal example of the use of the law we continue with the difference example.

$$
\begin{array}{ll}
\mathbf{F} \Rightarrow \mathit{diff}\ 1\,2 = -1 & \text{Bool Base Law} \\
\quad \text{Type Checking Proof Obligation} & \\
\quad (\mathit{diff}\ 1\,2 = -1) : \mathit{bool} & \text{Interpreter laws} \\
\quad = \mathcal{I}\text{“}\mathit{diff}\ 1\,2 = -1\text{”} : \mathit{bool} & \text{String Assoc.} \\
\quad = \mathcal{I}(\text{“}\mathit{diff}\ 1\,2\text{”}; \text{“}=\text{”}; \text{“}-1\text{”}) : \mathit{bool} & \text{Bool Equality} \\
\quad = \mathbf{T} & \\
= \mathbf{T} &
\end{array} \tag{44}
$$

As we can see in the example, since the interpreter unquotes expressions, using it in proofs is usually just the reverse process.

A. Implementation

In general, implementing laws that use the interpreter in a theorem prover is non-trivial. This is because it is difficult to determine if unification alone is sufficient to check if a law applies. We deliberately wrote two equality laws differently to illustrate a couple cases where this task can be made easy. If the only place the interpreter appears in a law is the expression $\mathcal{I}\,lang$ in the domain of a variable, it can be treated as a generic type. Type checking can be done by scanning to see that the interpreter does not appear in any instantiated expression with a generic type. In the case of the second law, instantiating the variables and parsing yields a valid expression without any further computation.

VI. Metalogical Reasoning Within the Logic

There are several benefits of defining the interpreter and using it to create laws. One such benefit is the creation of generic laws, where type-checking for variables is not necessary. The removal of type-checking is not only beneficial for simplicity, partiality, and efficiency, but some operators are meant to be truly generic. For example, the left operand of the set-membership operator ($\in$) can be any expression in the language, and set brackets can be placed around any expression. By including the interpreter in the logic these laws are expressed with full formality. For sets, an example would be

$$\forall A, B : \mathcal{I}\,lang \cdot (\{A\} = \{B\}) = (A = B) \tag{45}$$

Another benefit is compact laws. For example, we wish to define a generic symmetry law for natural arithmetic in our logic. If we had a prefix notation then we could have written the law as

$$\forall f : (+, \times, =) \cdot \forall a, b : nat \cdot = (f\,a\,b)(f\,b\,a) \tag{46}$$

Using the interpreter we can create a law in a similar fashion for non-prefix notation.

$$
\begin{aligned}
&\forall f : \text{“}+\text{”}, \text{“}\times\text{”}, \text{“}=\text{”} \cdot \forall a, b : lang \cdot \\
&\qquad \mathcal{I}\,(a, b) : nat \Rightarrow \mathcal{I}\,(a; f; b) = \mathcal{I}\,(b; f; a)
\end{aligned} \tag{47}
$$

This law can be made completely generic and include more than arithmetic operators. It even becomes simpler to write.

$$\forall f : \text{“}+\text{”}, \text{“}\times\text{”}, \text{“}\wedge\text{”}, \text{“}\vee\text{”}, \text{“}=\text{”} \cdot \forall a, b : lang \cdot \tag{48}$$
$$\mathcal{I}\,(a; f; b) = \mathcal{I}\,(b; f; a) \tag{49}$$

These sorts of laws allow us to capture an idea like associativity or commutativity in a compact way, and can be easily extended by concatenating to the operator text. Some further abbreviations can be particularly useful:

$$\forall v \cdot P \;=\; \forall v : \mathcal{I}\,lang \cdot P \tag{50}$$
$$\exists v \cdot P \;=\; \exists v : \mathcal{I}\,lang \cdot P \tag{51}$$
$$\Sigma v \cdot P \;=\; \Sigma v : \mathcal{I}\,lang \cdot P \tag{52}$$
$$\S v \cdot P \;=\; \S v : \mathcal{I}\,lang \cdot P \tag{53}$$
$$\langle v \to P \rangle \;=\; \langle v : \mathcal{I}\,lang \to P \rangle \tag{54}$$

It appears as if the language has unrestricted quantification, comprehension, and domain-less functions. This is useful for generic quantification and generic functions. Of course, the expressions in the domains of the quantifiers and function above must not include the interpreter.

One of the most useful features of the interpreter is reasoning about the syntactic structure of an expression without requiring a meta-logic. These laws include function application and several programming laws. Some laws have caveats, such as requiring that in some expressions certain variables or operators do not appear. For example, there is a quantifier law for $\forall$ that says if the variable a does not appear free in P then

$$(\forall a : D \cdot P) = P \tag{55}$$

We would like to formalize this caveat. It is straight forward to write a program that checks variable or operator appearance in a string (respecting scope). We formalize a specification of the “no free variable” requirement using the interpreter. For simplicity, assume that variables are single characters, and strings are not in the interpreted language. For a string P in our language and a variable named a we specify

$$
\begin{aligned}
&\exists i : (0, .. \leftrightarrow P) \cdot P_i = \text{“}a\text{”} \quad \wedge \\
&\quad \neg\exists s, t, D, pre, post : {*char} \cdot \\
&\qquad (pre; \text{“}\langle a : \text{”}; D; \text{“} \to \text{”}; s; P_i; t; \text{“}\rangle\text{”}; post) = P \\
&\quad\ \vee (pre; \text{“}\langle\text{”}; P_i; D; \text{“} \to \text{”}; s; \text{“}\rangle\text{”}; post) = P
\end{aligned} \tag{56}
$$

This specification says that a is free in P. The first part says that there is an index i in P at which a appears. The second part says that a is not local. Let $free$ denote this specification parameterized for an expression and a variable; $free$ “a” P

says that a is free in P. The caveat for the quantifier law is formalized as

$$\neg(free\,\text{“}a\text{”}\,P) \quad \Rightarrow \quad \mathcal{I}\,(\text{“}\forall a:\text{”}; D; \text{“}\cdot\text{”}; P) = \mathcal{I}\,P \quad (57)$$

In a similar manner we can avoid including axiom schemata in some theories and have just a single axiom. The notation allows us to refer to all variables in an expression.

VII. Fixed-Points

Quines are self-reproducing expressions; their interpretation is equal to themselves. Fixed-points of the interpreter are then Quines, formally satisfying

$$\mathcal{I}\,Q \;=\; Q \quad (58)$$

A Quine under this definition need not be a program. The following expression is a Quine [17]:

$$\text{“}\backslash\text{“}\backslash\text{“}[0;2{*}(0,..15)]\backslash\text{”}[0;2{*}(0,..15)]\text{”} \quad (59)$$

Of course, if we have $Q = \text{“}Q\text{”}$ then it is trivially a Quine, and therefore the definition of Quines is often restricted to expressions with no free variables.

VIII. Including the Interpreter

The definitions above exclude the interpreter itself from the interpreted language to maintain consistency. Gödel's First Incompleteness Theorem implies that it is not possible to define the interpreter to be both consistent and complete [19], [20]. A simpler proof of Gödel's theorem by [18] shows why a straight-forward inclusion of the interpreter by the laws

$$\text{“}\mathcal{I}\text{”}; lang \;:\; lang \quad (60)$$

$$\forall s : lang \cdot \mathcal{I}\,\text{“}\mathcal{I}\text{”}; s \;=\; \mathcal{I}\,s \quad (61)$$

is inconsistent. In that paper the interpreter was defined to be a mapping $lang \to bool$, which is not the case without definitions. Nonetheless, since both completeness and consistency cannot be achieved at once, any consistent logic that includes the interpreter will necessarily include strings that cannot be consistently interpreted, and hence be incomplete. Specifically, let rus be the expression $\{\S x : \mathcal{I}\,lang \cdot \neg(x \in x)\}$. Then rus cannot have its string representation interpreted consistently. The proof is as follows, and is similar to Russell's paradox.

$$\begin{aligned}
&rus \in rus && (62)\\
=\;& rus \in \{\S x : \mathcal{I}\,lang \cdot \neg(x \in x)\}\\
=\;& rus : (\S x : \mathcal{I}\,lang \cdot \neg(x \in x))\\
=\;& \neg(rus \in rus) \wedge rus : \mathcal{I}\,lang\\
=\;& \neg(rus \in rus) \wedge \{\S x : \mathcal{I}\,lang \cdot \neg(x \in x)\} : \mathcal{I}\,lang\\
=\;& \neg(rus \in rus)
\end{aligned}$$

Since $rus \in rus$ is boolean and an element, the proof above shows a contradiction in the logic. The interpreter is therefore incomplete for expressions such as rus, but all expressions for which the interpreter is incomplete include the interpreter.

However, as [18] also suggests, any logic can be completely described by another. This point is intuitively manifested in the fact that all expressions that cannot be interpreted include the interpreter itself. In a sense, we relegate all issues of partiality in our logic to involve only the interpreter.

However, we can weaken the restriction on the interpreter being excluded from the language. The motivation for including the interpreter is to reason about languages that allow this sort of self-reference. In practice, theorem provers such Coq [2] allow reflection as a proving technique. Reflection is a proof technique that allows a program (written in the functional language of Coq) to reason about expressions in Coq syntactically (at a meta-level). For example, a tactic for normalizing variable ordering in arithmetic equations would need to reason about expressions syntactically. It improves the performance of proof search considerably by eliminating the need for a number of applications of a commutative law.

A simple interpreter can be defined for arithmetic expressions in Coq in a straight-forward manner: the semantics of an expression would parallel its syntactic definition. However, such an interpreter must be well-typed, while the interpreter in this paper may fail to denote a value. It is non-trivial to define an interpreter in Coq whose interpreted language includes the interpreter itself.

We would like to use the interpreter as a simple way of reasoning about termination and consistency of definitions. The key insight is that a mathematical function disregards computation time.

The domain $xnat$ is the naturals extended with ∞. The domain of both $\mathcal{T}$ and $\mathcal{T}_{\mathcal{I}}$ is nat and their range is $xnat$. We define a parsing function from strings in the language to a tree data structure as follows:

$$\forall v : var, num, string, \text{“}\mathbf{T}\text{”}, \text{“}\mathbf{F}\text{”} \cdot \mathbf{parse}\,v = \mathbf{graft}\,v\,nil$$

$$\forall s,t : lang \cdot \forall v : var \cdot \forall bin : binops \cdot \forall uni : uniops \cdot$$

$$\begin{aligned}
&\mathbf{parse}\,uni; v && = \mathbf{graft}\,uni\,(\mathbf{parse}\,s)\\
&\mathbf{parse}\,s; bin; t && = \mathbf{graft}\,bin\,(\mathbf{parse}\,s); (\mathbf{parse}\,t)\\
&\mathbf{parse}\;\text{“(”}; s; \text{“)”} && = \mathbf{graft}\;\text{“()”}\;(\mathbf{parse}\,s)\\
&\mathbf{parse}\;\text{“}\langle\text{”}; v; \text{“:”}; s; \text{“}\to\text{”}; t; \text{“}\rangle\text{”} && = \mathbf{graft}\;\text{“}\langle\rangle\text{”}\;(\mathbf{parse}\,s); (\mathbf{parse}\,t)
\end{aligned}$$

We can measure interpretation time recursively by defining the following timing functions.

$$
\begin{aligned}
&\mathcal{T}\, nil = \mathcal{T}_{\mathcal{I}}\, nil = 0\\
&\mathcal{T}\, s = \mathcal{T}\,(\mathbf{parse}\, s)\\
&\mathcal{T}\,\mathbf{graft}\, op\, subtrees \quad = \quad \mathbf{if}\ op = \text{“}\mathcal{I}\text{”}\ \mathbf{then}\\
&\qquad\qquad 1 + \Sigma\mathcal{T}_{\mathcal{I}}\,(subtrees)\\
&\qquad \mathbf{else}\\
&\qquad\qquad \Sigma\mathcal{T}(subtrees)\\
&\mathcal{T}_{\mathcal{I}}\,\mathbf{graft}\, op\, subtrees \quad = \quad \mathbf{if}\ op = \text{“}\mathcal{I}\text{”}\ \mathbf{then}\\
&\qquad\qquad 1 + \Sigma\mathcal{T}_{\mathcal{I}}\,(subtrees)\\
&\qquad \mathbf{else\ if}\ op : var\\
&\qquad\qquad \mathcal{T}\,(\mathbf{parse}\,(\mathcal{I}\, op))\\
&\qquad \mathbf{else}\\
&\qquad\qquad \Sigma\mathcal{T}_{\mathcal{I}}(subtrees)
\end{aligned}
$$

This function is in a way parallel to how an interpretation works, except that it counts time. The time in question is the number of law applications needed to simplify an expression to have no interpreter symbol in it. At each "if"-statement the function checks for the occurrence of a certain piece of syntax, and the vertical ellipsis would include a similar check for the rest of the syntax. The special part of this function is when we see the interpreter symbol. If the interpreter was applied to a string representing a variable, and that variable's value is a string in the language, we recurse on its value. If the interpreter is applied to any other expression, we recurse on that expression's string representation. For example, if we have

$$Q = \text{“}\neg\mathcal{I}\, Q\text{”} \tag{63}$$

then we calculate

$$
\begin{aligned}
&\mathcal{T}\, Q \\
&= \mathcal{T}\,\text{“}\neg\mathcal{I}\, Q\text{”}\\
&= \mathcal{T}\,\text{“}\mathcal{I}\, Q\text{”}\\
&= \mathcal{T}_{\mathcal{I}}\,\text{“}Q\text{”}\\
&= 1 + \mathcal{T}\, Q
\end{aligned} \tag{64}
$$

and therefore $\mathcal{T} Q = \infty$ since $\mathcal{T} Q : xnat$. For any string that does not include the interpreter the time is linear in the size of the string; this can be proven by structural induction over *lang* if we add an induction axiom along with the construction axioms we defined earlier. We should only interpret an expression that includes the interpreter if the execution time of the interpretation is finite. If it is infinite or cannot be determined, then there is a potential for inconsistency had we decided to interpret it regardless. We can add the interpreter to the interpreted language as follows:

$$\forall s : lang \cdot \mathcal{T} s {<} \infty \ \Rightarrow\ \mathcal{I}\,\text{“}\mathcal{I}\text{”}; s = \mathcal{I}\, s \tag{65}$$

As an example of calculating with the interpreter, consider an expression normalizer $\mathcal{N}$ that linearizes an associative expression. For a sample input of "$a+((b+c)+d)$" it would output "$a+(b+(c+d))$". For the language $l_{\mathcal{N}}$ of expressions containing variables, brackets and plus, $\mathcal{N}$ is a total function. A partial specification of its behaviour is

$$\forall s : l_{\mathcal{N}} \cdot \mathcal{I}\,(\mathcal{N}\, s) \ = \ \mathcal{I}\, s \tag{66}$$

We want to normalize the expression $(a + \mathcal{I}\, t) + (b + c)$. The sub-part t of the expression that is input to $\mathcal{N}$ might be unknown, such as if it came from a stream communicating with a different process. However, if we know that $t : l\mathcal{N}$, and that the variables in string t do not contain a, b, c, then it is reasonable to prove that $(a+\mathcal{I}\, t)+(b+c) \ = \ a+(b+(c+\mathcal{I}\, t))$ using the definition of the normalizer. The calculation would be

$$
\begin{aligned}
&(a + \mathcal{I}\, t) + (b + c)\\
&= \mathcal{I}\,\text{“}(a + \mathcal{I}\, t) + (b + c)\text{”}\\
&= \mathcal{I}\,(\mathcal{N}\,\text{“}(a + \mathcal{I}\, t) + (b + c)\text{”})\\
&= \mathcal{I}\,\text{“}a + (b + (c + \mathcal{I}\, t))\text{”}\\
&= a + (b + (c + \mathcal{I}\, t))
\end{aligned} \tag{67}
$$

Although the proof appears simple, a proof obligation required to justify the steps is that the interpretation time of $\mathcal{I}\, t$ must be finite. We argue that this example is representative of real computations that can be reasoned about using the interpreter.

The requirement to prove finite interpretation time in order to evaluate the interpretation of a string in the language is similar to the concepts of partial and total correctness proofs in program theories [3], [15]. One is a proof about the result of execution, and the other about the execution time. Many programming theories require finding either a bounded-decreasing function of the input to a program, or fixed-points for loops [21] [3]. Other functional and proof languages restrict the language itself, often constructively. In effect, that is excluding those strings from the language that cannot be built constructively. Instead of interpretation time, we can define constructively the interpretable strings which include the interpreter. Let those strings be called $ilang$, whose definition parallels the definition of $lang$, except that for variables:

$$\forall v : var \cdot \mathcal{I}\, v : ilang \Rightarrow v : ilang \tag{68}$$

And then we add that

$$\text{“}\mathcal{I}\text{”}; ilang \ : \ ilang \tag{69}$$

$$\text{“}\mathcal{I}\text{”}; ilang \ : \ lang \tag{70}$$

We prove that for all strings in $ilang$ the interpretation time is finite by induction. The base case is strings in $terminal$, for which $\mathcal{T}$ is zero. For strings in unary, binary or bracketing operators $\mathcal{T}$ is the sum of the interpretation time of the operands. For strings pre-fixed with the interpreter, $\mathcal{T}$ is one plus $\mathcal{T}_{\mathcal{I}}$ of the operand, which by the induction hypothesis is finite. Finally, only variables whose interpretation is in $ilang$ may be included, so for $v : var$ we have $\mathcal{T}_{\mathcal{I}}\, v \ = \ \mathcal{T}\,(\mathcal{I}\, v)$

which is finite by the inductive hypothesis, which concludes the proof ■.

Therefore, no expressive power was gained with a restricted language as opposed to demanding proof of finite interpretation time. We argue that it is preferable not to restrict the language to finite interpretation time, because reasoning about strings with infinite interpretation time can be of the same use as reasoning about infinite computations. The antecedent requiring finite interpretation time is sufficient for consistency. However, not all strings in the language whose interpretation time is infinite cause an inconsistency.

In general, proving a finite execution time is the halting problem. When reasoning about logics it may be useful to include the interpreter in the interpreted language. For many practical purposes it can be left out.

IX. General Recursion in Functional Languages and Lazy Evaluation

The Trellys project [22][5] aims to create a functional language with general recursion and dependent types. The current approach is to separate the logical language from the computational language, because otherwise the result is inconsistency. The logical language and computational language share some parts, such as some shared data-types, but it is only the computational language which may have general unrestricted recursion. However, the interpreter can be applied to decouple timing from the language definition and allow the logic and computational language to be the same.

We claim that the interpreter can be used to simply and expressively reason about functional programming languages, and to define one with general recursion in a simple way. To do this, the interpretation of a functional program requires a finite execution time, and we decouple these two properties using the interpreter. Consider the following definition for the simple language of lambda calculus $\lambda lang$.

$$var : \lambda lang \tag{71}$$

$$\text{“(”}; \lambda lang; \text{“ ”}; \lambda lang; \text{“)”} \ : \ \lambda lang \tag{72}$$

$$\text{“}\lambda\text{”}; var; \text{“}\cdot\text{”}; \lambda lang \ : \ \lambda lang \tag{73}$$

Although consistent within its own domain, lambda calculus is not consistent when combined with other theories such as boolean algebra and arithmetic. However, even without requiring typed lambda terms, there are programming languages such as Python that implement consistently lambda terms within the programming language. This suggests that by restricting the evaluation of β-reductions to only those with finite execution time can resolve the inconsistency. For $v : var$ and $s, t : \lambda lang$ we define interpretation for lambda calculus as follows:

$$\mathcal{I}_\lambda \text{“}\lambda v \cdot\text{”}; s \ = \ \lambda v \cdot \mathcal{I}_\lambda\, s \tag{74}$$

$$\mathcal{I}_\lambda \text{“(”}; s; \text{“)”} \ = \ \mathcal{I}_\lambda\, s \tag{75}$$

$$\mathcal{T}_\lambda\,(s; \text{“ ”}; t) < \infty \ \Rightarrow \qquad \mathcal{I}_\lambda\, s; \text{“ ”}; t \ = \ (\mathcal{I}_\lambda\, s)\,(\mathcal{I}_\lambda\, t) \tag{76}$$

In the same way as before, variables are abbreviation of the interpretation of strings in var, so $\mathcal{I}_\lambda$ “a” is an abbreviation of a. We assume the existence of a function β that performs β-reductions on strings in $\lambda lang$. For $v : var$ and $s, t : \lambda lang$ the timing function for lambda calculus is:

$$\mathcal{T}_\lambda\, v \ = \ 0 \tag{77}$$

$$\mathcal{T}_\lambda \text{“}\lambda\text{”}; v; \text{“}\cdot\text{”}; s \ = \ \mathcal{T}_\lambda\, s \tag{78}$$

$$\mathcal{T}_\lambda \text{“(”}; s; \text{“ ”}; t; \text{“)”} \ = \ 1 + \mathcal{T}_\lambda\,(\beta\, s\, t) \tag{79}$$

The proof of inconsistency for untyped lambda calculus shows that the β-reduction of $(\lambda x \cdot \neg(x\ x))\ (\lambda x \cdot \neg(x\ x))$ is equal to its negation. However, the definition of $\mathcal{I}_\lambda$ requires finite interpretation time for β-reduction, which is not the case for this term:

$$\mathcal{T}_\lambda \text{“}((\lambda x \cdot \neg(x\ x))\ (\lambda x \cdot \neg(x\ x)))\text{”} \tag{80}$$

$$= 1 + \mathcal{T}_\lambda\, \beta\,(\text{“}\lambda x \cdot \neg(x\ x)\text{”}\ \text{“}\lambda x \cdot \neg(x\ x)\text{”}) \tag{81}$$

$$= 1 + \mathcal{T}_\lambda \text{“}\neg((\lambda x \cdot \neg(x\ x))\ (\lambda x \cdot \neg(x\ x)))\text{”} \tag{82}$$

$$= 1 + \mathcal{T}_\lambda \text{“}(\lambda x \cdot \neg(x\ x))\ (\lambda x \cdot \neg(x\ x))\text{”} \tag{83}$$

And therefore the interpretation time is equal to ∞ as before. We have decoupled interpretation from execution, and made the theory consistent. The interpretation time was mapped to the underlying computation model, albeit slightly abstracted: β-reduction costs time 1, and all else is free.

In general, for any logic whose inconsistency is due purely to operators that take infinite time to compute, it can be used as a consistent computation logic with no change. In essence, the computation logic of a theory makes operators incomplete for infinite executions. For this reason lambda calculus can be used as a computation logic with no change.

Furthermore, interpretation time allows a flexible timing policy. If instead the computation model is an Oracle Turing Machine, then the cost of β-reduction becomes finite even for infinite computations, and the theory becomes an inconsistent computation logic. The timing function can be further generalized to consider valuation of inputs. This is particularly useful for reasoning about lazy evaluation of functional programs, because a lazy language can use the exact same logic as its non-lazy counter-part with the only difference being their interpretation time. A lazy language can allow for infinite datatypes such as infinite lists, or inductively defined infinite datatypes. Consider the following definition of an infinite list of naturals:

$$natlist \ = \ 0; (1 + natlist) \tag{84}$$

which can be equivalently defined without explicit recursion using a Y-combinator as

$$natlist = (\lambda f \cdot (f\, f))\ (\lambda s \cdot 0; (1 + (s\, s))) \tag{85}$$

and that we wish to evaluate $natlist_{42}$. As currently defined, $\mathcal{T}_\lambda$ is a lazy timing policy, and $\mathcal{T}_\lambda\,(natlist_{42}) \ = \ 42$. An eager timing policy results in infinite execution time, and is achieved by the simple change to the timing function:

$$\mathcal{T}_\lambda \text{“(”}; s; \text{“ ”}; t; \text{“)”} \ = \ 1 + (\beta\,(\mathcal{T}_\lambda\, s)\,(\mathcal{T}_\lambda\, t)) \tag{86}$$

X. Proof of Consistency

To prove the interpreter consistent we will find a model in set theory. Characters are implemented as natural numbers, having

$$\text{“0”} = 0,\ \ldots \text{“9”} = 9, \text{“}a\text{”} = 10,\ \ldots \text{“}z\text{”} = 25,\ \ldots \tag{87}$$

Strings are implemented as ordered pairs in the standard way.

$$a; b = \{\{a\}, \{a, b\}\} \tag{88}$$

The interpreter is a mapping from the set of all strings in our language $lang$ to the class of all sets. $\mathcal{I} \subseteq lang \times Sets$. It is assumed that all other theories (functions, boolean algebra, numbers) are implemented in set theory in the standard way. For this reason partial functions might be implemented using another special value that all remaining domain elements will be mapped to. We will not delve into the implementation of functions and other theories, since once they are implemented in set theory, they are included in the class $Sets$.

We must prove that there exists a function $\mathcal{I}$ such that the interpreter axioms are true. The recursion theorem will be used to prove this [9]. The theorem states that given a set X, an element a of X, and a function $f : X \to X$ there exists a unique function F such that

$$F\,0 = a \tag{89}$$
$$\forall n : nat \cdot F\,(n+1) = f(F\,n) \tag{90}$$

Since $\mathcal{I} \subseteq lang \times Sets$ it is necessary to first find a function from $lang$ to the naturals; this is an enumeration of the strings in $lang$. Let $charNum$ be the total number of characters in $char$. Character string comparison for strings s, t is defined as

$$(s > t) = strNum(s) > strNum(t) \tag{91}$$
$$strNum = \langle S : {*char} \to \textbf{if}\ S = nil\ \textbf{then}\ 0 \quad \textbf{else}\ S_0 + charNum \times S_{1,..\leftrightarrow S}\ \textbf{fi}\rangle \tag{92}$$

The enumeration function $enum$ of strings in $lang$ is defined as

$$enum = (g^{-1}) \tag{93}$$
$$g = \langle n : nat \to \textbf{if}\ n = 0\ \textbf{then}\ (MIN\, s : lang \cdot s) \quad \textbf{else}\ (MIN\, s : (\S t : lang \cdot t > g(n-1)) \cdot s)\ \textbf{fi}\rangle \tag{94}$$

The function $strNum$ assigns a unique number to each string. Some character strings are not in $lang$, and we desire an enumeration free from gaps. The function g assigns a unique string in $lang$ to each natural as follows: zero is mapped to the first string in the language, and each subsequent number is mapped to the next smallest string. Since g is one-to-one, we define $enum$ as its inverse. We define function F for a given state in the model with finite single-character variables as

$$F\,0 = \{0; 0\} \tag{95}$$
$$\vdots$$
$$F\,9 = \{9; 9\} \cup F\,8$$

For all $s : char$ let $m = enum(\text{“}\backslash\text{“”}; s; \text{“}\backslash\text{””})$ in

$$F\,m = \{m; s\} \cup F\,(m-1) \tag{96}$$

$$F\,(n+1) = \{(n+1); (H\,(n+1)\,(F\,n))\} \cup F\,n \tag{97}$$

(H is defined below)

At each argument n function F is a mapping of all previous numbers to their corresponding expressions, in addition to the current one. The base elements are the variables, numbers and strings. Function H constructs expressions using the operators in our language from previous expressions. It is defined as follows.

$$\begin{aligned}
&H\,k\,I = \\
&\{S : Sets | \exists n, m : \mathbf{dom}(I) \cdot g\,k = (g\,n); \text{“}+\text{”}; (g\,m) \\
&\qquad \wedge S = I\,n + I\,m\} \cup \\
&\{S : Sets | \exists n, m : \mathbf{dom}(I) \cdot g\,k = (g\,n); \text{“}\wedge\text{”}; (g\,m) \\
&\qquad \wedge S = I\,n \wedge I\,m\} \cup \\
&\{S : Sets | \exists n : \mathbf{dom}(I) \cdot g\,k = \text{“}\neg\text{”}; (g\,n) \wedge S = \neg I\,n\} \cup \\
&\vdots
\end{aligned} \tag{98}$$

The vertical ellipsis represents a similar treatment for other operators and is used to save space. Since g is one-to-one, only a single set in this union will have an element in it. In other words, each number is mapped to a single expression (but not vice-versa). Finally, the interpreter is implemented as follows.

$$\mathcal{I} = \langle s : lang \to (F\ (enum\, s))\,(enum\, s)\rangle \tag{99}$$

XI. Conclusion

We have presented the formalism of an expression interpreter for the purpose of encoding meta-logic within a logic. This allows effective reasoning with partial terms and about functional languages. Our technique requires no separate meta-logic, and we believe that our encoding of expressions as character strings is simple and transparent. The use of the interpreter allows proofs with partial terms to proceed in a fully formal fashion classically; that is, with just the standard boolean algebra. We show how the interpreter can be used to create generic and compact laws, which also allow syntactic reasoning about expressions. We also argue that the incorporation of the interpreter in theorem provers is reasonable, since the parsing that is required for its use is an efficient linear-time algorithm. We show how the interpreter can be used to implement a logic for functional programming that allows general recursion while maintaining consistency.

The encoding permits a flexible timing policy of execution, which also allows reasoning about lazy evaluation.

REFERENCES

[1] L. Naiman, "Using an expression interpreter to reason with partial terms," in *COMPUTATION TOOLS 2013: the fourth international conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking*, ser. IARIA '11, 2011, pp. 37–43.

[2] The Coq development team. (2004) The coq proof assistant reference manual. LogiCal Project. Version 8.0. [Online]. Available: http://coq.inria.fr. Access: 2013-01-20.

[3] C. C. Morgan, *Programming from specifications, 2nd Edition*. Upper Saddle River, NJ, USA: Prentice Hall, 1994.

[4] C. B. Jones, *Systematic Software Development Using VDM (2Nd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990.

[5] G. Kimmell, A. Stump, H. D. Eades, III, P. Fu, T. Sheard, S. Weirich, C. Casinghino, V. Sjöberg, N. Collins, and K. Y. Ahn, "Equational reasoning about programs with general recursion and call-by-value semantics," in *Proceedings of the Sixth Workshop on Programming Languages Meets Program Verification*, ser. PLPV '12. New York, NY, USA: ACM, 2012, pp. 15–26.

[6] C. B. Jones and C. A. Middelburg, "A typed logic of partial functions reconstructed classically," *ACTA*, vol. 31, no. 5, pp. 399–430, 1994.

[7] C. B. Jones, M. J. Lovert, and L. J. Steggles, "A semantic analysis of logics that cope with partial terms," in *ABZ*, ser. LNCS, J. Derrick, J. A. Fitzgerald, S. Gnesi, S. Khurshid, M. Leuschel, S. Reeves, and E. Riccobene, Eds., vol. 7316. Springer, 2012, pp. 252–265.

[8] C. B. Jones, "Partial functions and logics: A warning," *IPL*, vol. 54, no. 2, pp. 65–67, 1995.

[9] W. Just and M. Weese, *Discovering Modern Set Theory. I.* American Mathematical Society, 1996, vol. 8.

[10] M. Beeson, *Foundations of Constructive Mathematics*. New York, NY, USA: Springer-Verlag, 1985.

[11] ——, "Lambda logic," in *Automated Reasoning: Second International Joint Conference, IJCAR 2004*. Springer, 2004, pp. 4–8.

[12] R. F. Stärk, "Why the constant 'undefined'? logics of partial terms for strict and non-strict functional programming languages," *J. Funct. Program.*, vol. 8, no. 2, pp. 97–129, 1998.

[13] R. D. Gumb, "The lazy logic of partial terms," *JSYML*, vol. 67, no. 3, pp. 1065–1077, 2002.

[14] M. J. C. Gordon, R. Milner, and C. P. Wadsworth, *Edinburgh LCF*, ser. Lecture Notes in Computer Science. Springer, 1979, vol. 78.

[15] D. Gries, *The Science of Programming*. New York: Springer-Verlang, 1981.

[16] J. Woodcock and V. Bandur, "Unifying theories of undefinedness in utp," in *UTP*, ser. LNCS, B. Wolff, M.-C. Gaudel, and A. Feliachi, Eds., vol. 7681. Springer, 2012, pp. 1–22.

[17] E. C. R. Hehner, *A Practical Theory of Programming*. New York: Springer, 1993. [Online]. Available: http://www.cs.toronto.edu/ hehner/aPToP/. Access: 2014-05-27

[18] ——, "Beautifying gödel," pp. 163–172, 1990.

[19] K. Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme," *Monatshefte für Mathematik und Physik*, vol. 38, no. 1, pp. 173–198, 1931.

[20] R. Zach, "Kurt gödel and computability theory," in *Logical Approaches to Computational Barriers*, ser. Lecture Notes in Computer Science, A. Beckmann, U. Berger, B. Löwe, and J. Tucker, Eds. Springer Berlin Heidelberg, 2006, vol. 3988, pp. 575–583.

[21] C. A. R. Hoare and J. He, *Unifying Theories of Programming*. New York: Prentice Hall, 1998.

[22] V. Sjöberg, C. Casinghino, K. Y. Ahn, N. Collins, H. D. E. III, P. Fu, G. Kimmell, T. Sheard, A. Stump, and S. Weirich, "Irrelevance, heterogeneous equality, and call-by-value dependent type systems," in *MSFP*, 2012, pp. 112–162.

A Decision-making Support System for Land Use Estimation Based on a New Anthropentropy Predictive Model for Environmental Preservation

Theory, Model, and Web–based Implementation

Maria Grazia Albanesi
Dept. of Electrical, Computer and Biomedical Engineering
University of Pavia
Pavia, Italy
mariagrazia.albanesi@unipv.it

Roberto Albanesi
Thea s.r.l.
Pavia, Italy
ufficiostampa@albanesi.it

***Abstract*—This paper describes a new decision-making support system, which is able to estimate the future impact on the environment of new planned (but not yet built) urban settlements and/or communication roads. The challenging addressed problem is to decide if, according to a quantitative indicator, the creation of new human anthropic areas is compatible with a sustainable land use control, for an efficient environment preservation. The core of the system is a predictive model, which is initially trained by selected worst stressing cases. Some modifications to classical computer vision morphological operators are proposed and applied to standard Google Earth satellite maps, according to the User Generated Content paradigm. The model updates the previously defined indicator of Anthropentropy Factor, by producing a novel indicator of higher level (indicator of type C, or performance indicator, according to European Environmental Agency classification). The paper describes this important theoretical improvement, the model architecture, the new customized computer vision functions, and the prototype of a web-based implementation of the decision-making support system, with visual and numerical results of some significant cases.**

Keywords-Land use, urban sprawl, anthropentropy factor, decision-making support system, predictive model, morphological operators, web-based system, UGC.

I. INTRODUCTION

This paper describes the continuation and the extension of the ACI project [1]: both theory and implementation have been improved to meet more complex problems. The challenging task is the same: to give computer-based algorithms and methodologies to help the environmental preservation, specifically in the field of land use limitation against the threat of an inappropriate, out of control, urban sprawl. Land use is just one of the aspects of environmental protection, but surprisingly it is the one where few results have been reached: since 2006, European Environment Agency pointed out this fact, by calling urban sprawl "the ignored challenge" [2]. Even if the problem is well known, solutions are far from being proposed, accepted and adopted. Data are self-explaining: annual land take in 36 European countries was 111,788 ha/year in the period 2000-2006 [3], with sensible differences among countries: in the worst cases, annual land take increased by 9 %. In Italy, the situation is particularly evident, with an increment of land use of 6.3%, for the period 1956-2006 [4]. Recently, these data about Italian territory have been updated [5], with an esteemed growth of land take of 70 h a day. The most relevant consequence of land use is soil sealing: soil is modified, for the presence of asphalt, concrete and other human artifacts, and this fact prejudices some vital functions of the ecosystems, causes territory fragmentation and it is a serious threat to biodiversity. For this reason, the European Environment Agency classified the land use and biodiversity in the same policy target and objective [6], the last one, called "Biodiversity and land use". One of the more challenging task of computer science community is to give instruments to help governments and citizens to pay more attention to land use impact on environment and quality of life. The main effort of researchers is twofold: to define indicators for land use computation and to describe the current situation of the territory. These two problems have been addressed in the previous research [1], but this is not sufficient for a long-term policy of land use control, as pointed out in the middle term policy targets of 2030 [6]. In fact, the most challenging problem is to prevent land take, i.e., to give tools and methodology to predict the future anthropic expansion. The instant land take is dramatic in Italy [5]: every second, 8 square meters of green land are engulfed by new human settlements, industries, roads, intensive farming or touristic resorts. For this reason, starting from the indicator *Anthropentropy Factor* [1], a new decision-making support system is here proposed and described: the distinctive idea is to model the expansion of existing and future planned (but not yet built) anthropic places in a given territory. The model defines different classes of anthropic places and, for each of them, estimates the growth of their areas on the basis of initial assumptions and parameters fixed in the training phase. The model is dynamic, as the expansion is a function of time, with a time frame until 2050. For this reason, we called the model Dynamic ANTHropentropy Expansion model (i.e., DANTHE model), while we refer to the entire project as the DANTHE project. At the end of the expansion, the decision-making support system determines if the planned settlement is sustainable or not, according to a new metric of environmental preservation.

The rest of this paper is organized as follows. Section II describes the addressed problem and related work in literature. Section III describes the theoretical innovations introduced by the DANTHE project and the new dynamic indicator of land use. Section IV addresses the architecture and functions of the system. Section V describes the predictive model, and gives details on the new proposed morphological operators. Section VI describes the applications of the assumptions of the predictive model on some significant case studies and present the result on a real case on the Italian territory. Conclusion and considerations about future work close the article.

II. Related Work

The challenge is to compute, in an automatic or semiautomatic way, the land use, according to some indicator which represents numerically this concept. Before discussing about related work in literature, a brief summary of the terminology and definitions can be useful (in italics, the basic terms and their meaning adopted in the project).

A. *Terminology*

Land use can be calculated in different ways, and also its significance is often a source of misunderstanding. In its wider, and correct sense, land use is the classification, within a limited territory, of the areas of *anthropic places*, i.e., places occupied by humans (for their life, economic and productive activities), and of the areas of wild nature. A simplified list of anthropic places can be the following: buildings, such as housing, workplaces, schools, hospitals, paved roads, railways, and places of intensive agriculture. In all these anthropic places, the human presence is fairly continuous, and has effectively ousted the wild.

A correlated term is *land take*, which expresses the variation of the land use over time, generally referred to a specific time period (e.g., one year for annual land take).

Another term, whose meaning is often confused with that of land use, is *urban sprawl*. European Environment Agency defines the term urban sprawl as the physical pattern of low-density expansion of large urban areas [6]. In their expansion, the urban areas penetrate and destroy the surrounding agricultural areas. Sprawl is the leading edge of urban growth and this phenomenon usually implies little planning control. This consideration is particularly relevant and has become the starting point of the DANTHE project: increasing control over urban sprawl is mandatory for a policy of biodiversity preservation. If we had a tool to decide if a future configuration of an urban area will be sustainable over time and compatible with an intelligent policy of environmental preservation, this tool could be used in a simulation to study future urban expansions in a *scenario* of *What if?*. This could help local government in planning the annual Territory Government Plan, where new urban expansions are decided. This is exactly the goal of the decision-making support system DANTHE: to avoid the expansion of urban areas in the form of patchy, scattered development, with a high tendency for discontinuity. The consequences of discontinuity of urban sprawl will be further analyzed later in this section, when the problem of fragmentation will be discussed.

Unfortunately, urban sprawl is not the only accountable for land take: in fact, intensive farming and tourism, especially for coastlines, contribute to a relevant decrease of wild nature. Coasts are being urbanized at high rate, and are becoming twist together with the hinterland and more dependent on tourism and secondary homes. Economic changes support this evolution: as a consequence, also rural and coastline little villages are being growing according to an unplanned incremental urban development, exactly as big cities. This phenomenon has been confirmed also by the previous results of the ACI project [1], where land exploitations reaches worrying results also in coastline and rural regions (see Section II.B for new updated results of the ACI project regarding coastline regions.) For all these reasons, we are interested to every kind of anthropic places, not only urban settlements, but also communication lines (roads, airports, stations), areas for services, productive activities and recreational purpose settlements. While in the ACI project [1] we considered only four classes of anthropic places, in the DANTHE project the number of classes rises to 12; they are listed in Table I. The reasons of this new classification will be clear after the discussion of the basic assumptions and parameters of the model (Section V).

B. *Indicators for land use: state of the art*

In literature, few contributions refer to the problem of defining meaningful indicators to express the concept of land use. Several studies try to investigate the relationship among land use and other aspects of environmental degradation, for example water and air pollution. Land quality indicators [7] are currently under study, but their attention focuses not only on soil, but also on the complex intermingled relationship among terrain, water and biotic resources that provide the basis for land use. Land quality indicators always relate to agricultural areas and forestry, because they are more interested in the effects of land degradation over social and economic aspects of food production [8]. Therefore, these indicators express only a limited part of the general problem of land use.

Other approaches, such as bio indicators (populations of ants [9] and bryophytes [10]), are not able to decorrelate the land use from other aspects of environmental preservation, such as agro-biodiversity, water and soil contamination and pollution. Until now, it seems that the more interesting indicators for land use are the ones related

TABLE I. CLASSES OF ANTHROPIC PLACES, ACCORDING TO THE DANTHE MODEL.

Class Number	Classes of Anthropic Places	
	Class Name	***Some Examples***
1	Slow-growing settlements	schools, hotels, cemeteries, recreational small settlements, small shopping centers, including small parking lots
2	Fast-growing settlements	Houses (villas, cottages, mansions, possibly including small parking lots)
3	Commercial centers	Medium-large shopping centers, trade centers, malls
4	Industrial areas	Factories, industrial warehouses, logistics centers
5	Slow-growing areas of service production, venues for sport and health.	Business hubs, sports, recreational and health centers, waste treatment sites, energy production plants
6	Fast-growing areas of service production, venues for sport and health.	Stadiums, sports arenas, zoos, campuses, touristic resorts.
7	Airports and heliports	
8	Exhibition grounds	Venues for shows and trade fairs
9	Fast-growing roads	Highways or provincial roads
10	Slow-growing roads	Ring roads, railway lines, underground
11	Highways	
12	Stations	Bus, train stations

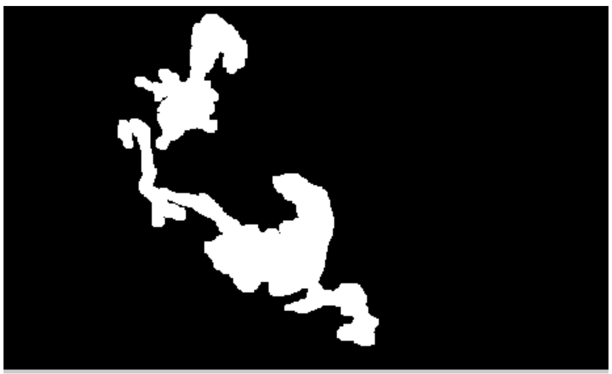
to area extensions: if we consider the definition of land use, the simplest indicator is the ratio between the area of all the anthropic places and the area of the territory under analysis. Unfortunately, this indicator, even if it is widely adopted [2-5], is not able to understand some crucial aspects for environment and biodiversity preservation. In particular, the simple numerical ratio cannot express the problem of fragmentation of wild areas. Fragmentation is caused by the disordered expansion of anthropic places, where the incremental areas (due to new human settlements) are distributed in the territory in such a way that new areas are not contiguous to existing ones. This is potentially a great drawback, because it increases fragmentation. The shape of wild land areas is important to assure a proper habitat for wild animal species: the fragmentation of the territory contributes greatly to limit a fundamental environmental aspect: biodiversity. In fact, the UN Convention on Biological Diversity [11] considers fragmentation as a major threat to habitats and species survival, because it causes insuperable barriers to the wandering and spreading of animals. We tried to overcome this problem by proposing a new indicator [1], called *Antropentropy Factor* (in the following, *AF*). Here, we recall the basic definition and concepts which are essential to understand the dynamic model DANTHE (for details on the properties of the *AF* indicator and its application to Italian territory, see [1]).

Anthropentropy is a neologism derived from the Greek term *Anthropos* (*Άνθρωπος*) = man, and *entropy*. In thermodynamics, entropy is the well-known measure of disorder of a system. The term anthropentropy puts in evidence the “disorder” introduced in a virgin, wild environment by the presence and disturbance of human beings. Land use is expressed by labeling a geographic area with a new indicator, called *Anthropentropy Factor*, which expresses in an absolute, continuous scale (from 0 to 1) the degree of penetration of human settlement in the environment. The *AF* indicator does not only computed the percentage of land occupied by human activities and urban expansion, but also it takes into consideration the *shape* of the areas subtracted to nature. In fact, the algorithm performs, before the final computation of area ratio, a morphological dilation [12], i.e., a geometrical enlargement of the anthropic regions, in which the four-connectivity contiguity of the areas is taken into consideration. In this way, also shapes and relative positions of the anthropic regions are determinant to the computation of *AF*, thus incorporating in the indicator the concept of fragmentation [1].

In order to compute the *AF*, let us consider a geographic region which can be bounded by identified borders in a proper scaled map (e.g., a municipality, a district or a county), and let define *S* its area (in square kilometers). The algorithm for *AF* computation proceeds to the identification of all the sub-regions occupied by the anthropic places of Table I. Each sub-region contains at least one of the anthropic places listed in Table I. Each area occupied by anthropic places is enlarged by the morphological operator of dilation (along the two Cartesian dimensions X and Y) with a factor of "buffering" (radius of the circular dilation) of 50 meters, to give rise to anthropic sub-regions. The choice of a 50 meters has been discussed in [1], and it seems a good compromise between a too restrictive and a too permissive limit. We define the union of all the anthropic sub-regions as *Death Zone* of the region. Let define *DA* as the area (in square kilometers) of the Death Zone. In Fig. 1a, the map of the anthropic places for the island Vulcano (Aeolian Islands, Sicily, Italy) is shown. In Fig. 1b, the corresponding Death Zone, after the morphological dilation, is shown.

We define a *neutral sub-region* as a part of the territory containing at least one of the following elements: (a) inland waters, (e.g., lakes or lagoons) extending more than two square kilometers (according to the limit of the Italian administrative coast boundary) and (b) lands located more than 3,000 m above sea level. The union of all the neutral sub-regions (if present), correspond to the *Neutral Zone*. Let define *NA* as the area (in square kilometers) of the Neutral Zone.

(a)



(b)

Figure 1. A step of the *AF* computation for the island Vulcano (Aeolian Islands, Sicily, Italy): (a) original map of the anthropic places (in red) (b) Corresponding Death Zone after dilation (in white).

If the geographic region does not contain at least one neutral sub-region, *NA* is set to 0. It is important to consider the Neutral Zone because it represents the regions where anthropic places are not possible, and its area has to be subtracted to the area *S* of the region, otherwise the land use becomes underestimated. Now, we have all the elements to define the *Anthropentropy Factor AF* as the ratio:

$$AF = DA / (S - NA) \quad (1)$$

The *AF* expresses the land use as a fractional number, between 0 (completely uninhabited regions without human active settlements and only wild nature, DA = 0) and 1 (fully populated regions, the Death Zone completely occupies the territory, but for the Neutral Zone (if any, where human settlements are not possible.) In (1) the special case of $NA = S$ is not admissible, as it would mean that the entire geographic area is occupied by water or is above 3,000 m above the sea, thus it is not suitable to the presence of human beings and the *FA* indicator becomes meaningless.

After expressing the value of *AF* for a given region, a metric is necessary to give a value to the indicator, and to link the numbers to a qualitative assessment of the environment and of the quality of life. We have chosen the following metric [1]: if *AF* is between 0 and 0.2, the region is considered at a very low level of anthropentropy (the ideal condition for nature and human beings). If *AF* is between 0.2 and 0.4, the situation is still good, but the region is associated to a worrying level of anthropentropy. This type of area have to be monitored, in time, to control its evolution, which potentially might reach undesired higher levels. If *AF* is between 0.4 and 0.6, the region is labeled with a serious level of anthropentropy. In these areas, the presence of humans negatively impacts on environment. If *AF* is between 0.6 and 1, the region is considered with a very serious level of anthropentropy, at such a point that an irreversible environmental degradation has been reached. The increasing levels of anthropentropy are represented visually on maps of the territory by coloring the regions into varying levels of green, yellow, red, violet and black. In the ACI Project [1], our reference geographic regions are the Italian municipalities, divided in administrative regions. For each region, it is possible to generate the corresponding *Anthropentropy Map*. In Fig. 2 and Fig. 3, the Anthropentropy Maps of regions Liguria and Puglia are shown, respectively. These maps have been processed and generated after the publication of the paper on the ACI Project [1], therefore, they can be considered as a new update and a completion of the previous results.

Obviously, in order to compute *AF*, or any other type of indicator based on area computation, it is mandatory to have e description of land occupation. For example, Fig. 2 and Fig. 3 are the result of the *AF* computation on the land cover description of the Corine Land Cover (CLC) project [13]. As already pointed out [1], Corine Land Cover data are not available for the whole territory, so other solutions have to be identified. One promising approach is to use remote sensing and satellite image data [14]: color or multi-spectral image processing primitives and classification algorithms are currently being investigated in order to define the land use. However, the main limitation, so far, are the difficulties in describing the whole territory, without a class of "unclassified regions"; in fact, if the classification algorithm fails in some part of the map, it is impossible to compute a precise value of land use, as the "unclassified region" cannot be attributed either to anthropic places nor to virgin and wild natural areas. The full automatic remote sensing approach seems to be more useful to detect changes [15] in the land use of a particular regions, for successive acquisition and differences, rather than to obtain a precise value for an indicator.

In literature, the most similar approach [16] to that of the DANTHE project refers only to urban sprawl: it has been tested for a specific geographic area of China (Jinan City). Moreover, the expansion of the urban area is modeled regardless of what there is around the same area, as if the growth was a context independent phenomenon. Instead, the DANTHE project overcomes these limitations and propose different ways of expansion of anthropic areas, depending on the type of the places, on their dimensions, and, especially, on neighboring areas, which may influence the growth in time. For this reason, the goal of the DANTHE project research has slightly changed, if compared to the ACI project [1]: we are not only interested to the value of land use indicator for a given region, computed according the existing, actual situation. Rather, we are interested in predicting the future value of *AF*, after a certain period of time, by taking in consideration new settlements in the territory, which have been planned and proposed but not yet built. We called this shift of goal “from a static to a dynamic dimension of *AF*”, and it corresponds to the main theoretic novelty of the DANTHE project; it can be described, according to the European Environment Agency, in terms of *type* of *environmental indicators*.

III. TOWARD A HIGHER LEVEL OF INDICATOR

According to the European Environment Agency [17], the environmental indicators can be classified in four types: descriptive indicators (type A), performance indicators (type B), efficiency indicators (type C), and Total Welfare indicators (type D). This classification holds for every kind of impact on the environment of the human activities, not only for the problem of land use.

The typology of environmental indicators refers to the DPSIR framework [17], where the complex interactions among the different human activities and the environment are described as a chain of causes-effects. The framework (Fig. 4) distinguishes driving forces (D), pressures (P), states (S), impacts (I) and responses (R). According to this framework, driving forces (mainly generated by social and economic motivations), generate pressures on the environment (usually negative pressures, such as pollution); as a consequence, the state of the environment changes. In turn, the state changes have a negative impact both on humans and eco-systems. The impact should generate a response which try to improve situations or remove negative impacts. The response should act on the driving forces which started the chain, in order to improve the state of the environment and the quality of life, by imposing a sort of virtuous feedback.

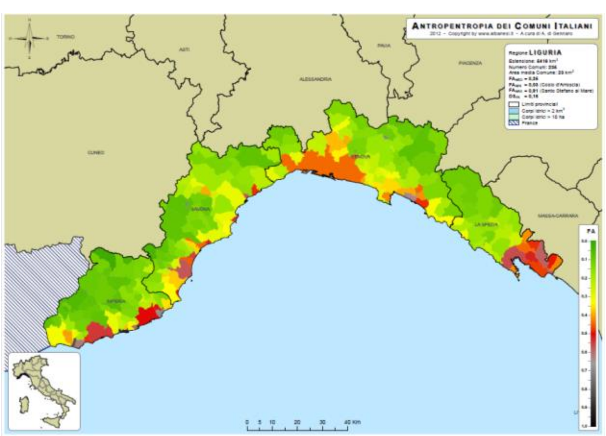


Figure 2. Anthropentropy Map of Region Liguria, Italy (land area: 5416 square kilometers, number of municipalities: 235).

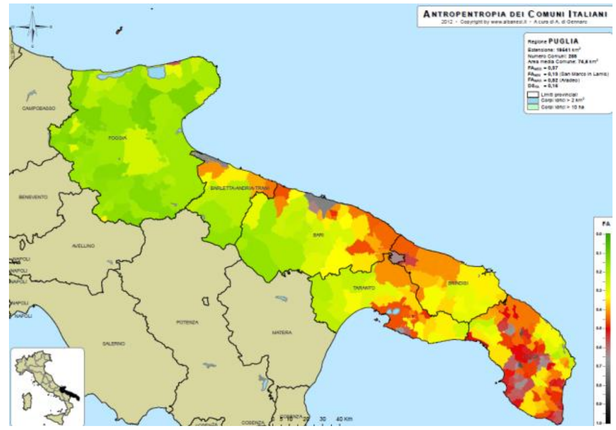


Figure 3. Anthropentropy Map of Region Puglia, Italy (land area: 19541 square kilometers, number of municipalities: 256).

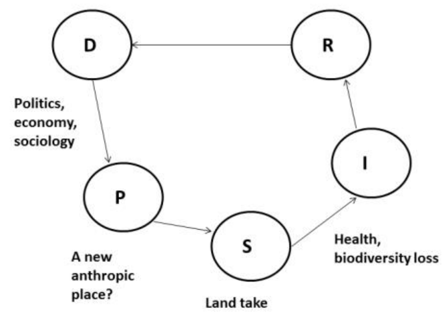


Figure 4. An example of application of the DPSIR framework to the problem of land use (D: driving forces, P: pressure, S: state, I: impact, R: response).

According to this model, indicators are well-defined if they are able to describe one or more of the links among the several actors of the framework (driving forces, pressures, states, impacts and responses). In order to apply the DPSIR framework to the problem of land use, one of the possible scenario can be the following: in a given territory, population growth (D, from a social point of view) leads to increasing demand for land use. Without any constraint imposed by policies of environmental sustainability (D, from a political point of view), this may results in an actual request of immoderate land use (P), which causes a significant change of state (S, e.g., degradation of soil quality, increase of greenhouse gas emissions, air and noise pollution). Thus, after a certain period of time, a large value of land take causes adverse impacts and negative effects (I) on the quality of life for people living in the territory, not only on animals and plants. In this scenario, the demographic growth and the absence of good environmental policies are the driving forces (D), the land take is the pressure (P), the degradation of soil and air are the changes on the state (S), and their consequences on health and quality of life are the impact (I). Fig. 4 shows this example of DPSIR for land use problem. The example underlines the greatest challenge: what about the response component of the framework? In order to close the chain and to implement the virtuous feedback, it is necessary to define the correct response and to investigate its effect on driving forces.

According to this framework, environmental indicators are classified by an increasing level of complexity:

- Type A indicators: these indicators describe the current, actual situation of a territory, by referring to a specific part of the DPSIR framework. For example, the Anthropentropy Factor (1) is a state indicator of type A, because it describes land use, measured according a precise algorithm/formula. State indicators give a description of the quantity and/or quality or some physical, biological or chemical aspects of the state of the environment. Other examples of state indicators are the concentration of toxic elements in lakes or the level of noise in a certain area.
- Type B indicators (performance indicators): type A indicators describe the situation as it is, without any reference to how the situation should be, in an optimum or near-optimum condition. In contrast, performance indicators compare the physical, biological, or chemical conditions to a specific set of reference conditions. The anthropentropy metric we have set in the ACI project [1] is an example of indicator of type B, as we compare the *AF* value to a set of intervals, where only the first one is desirable, the second one is near-optimum, and so on. The Anthropentropy Maps (Fig. 2 and Fig. 3) are the visual representation of a type B indicator for land use (*AF* definition *and* metric).
- Type C indicator (efficiency): type A and B indicators consider only some aspects of the DPSIR framework. However, it is desirable to create higher level indicators, which describe how, by acting on response, it is possible to improve the environmental preservation. A type C indicator necessarily is a function of time, as it answers the question “is the situation improving”? [17]. Most of the time, performance indicators take into account economic or social aspects to find out if, given a predetermined time period, the indicator shows, in its time evolution, that the environmental situation has (hopefully) improved.
- There is also a last type of indicator (type D, or total welfare indicator), which should answer the question: “are we on whole better off?”. However, to the authors' own admission, find an indicator of overall sustainability (i.e., which considers all aspects of environmental degradation) is a very ambitious goal and these type of indicators are not further described and investigated [17].

Unfortunately, type B and C indicators are very rare in some European countries, including Italy (for details, refer to Fig. 11 of the report [17] of the European Environment Agency.) For all these reasons, we have been strongly motivated by the main theoretical improvement of the DANTHE project: to evolve from a type B indicator (*AF* + metric = Anthropentropy Maps) to a type C indicator. The first step is to introduce the variable time: let define *AF(t)* the *AF* computed at a generic instant *t*, where *t* is expressed in years and its range is the discrete interval [2014:2050]. The upper limit 2050 is the same of the policy target of the European Environment Agency [6]. The second step is to define what we mean by *improved situation* (referring to land use); in fact, this is a concept which is implicit in a type C indicator. However, here there is a rub: for its specific nature, land use is, in some extents, an irreversible phenomena. In fact, it is unrealistic to think to act on drive forces to decrease land use, unless you destroy existing human settlements, but we does not take into consideration earthquakes or similar events! Land take has to be limited in the next years, but most likely it will never be negative; even in the most ambitious goals of the European Union environmental policy targets and objectives for years 2010–2050 [6], there are only partial desirable results, as to halt global forest cover loss (by 2030) and the net land take only for a limited subset of human settlements, i.e., for housing, industry, roads and recreational purposes (by 2050). Also the European Commission's roadmap to an efficient manage of resources [18] introduces the idea of “no net land take by 2050”. Also from the most optimistic point of view, the *AF* can only be constant (in the case of land take equal to zero) but, in more realistic situations, it increases. The model is more interesting (and useful) in the undesirable case in which the *AF* is not constant. For this reason, we can define a satisfactory improvement if the *AF* is constant or it increases, over time, with a growth rate such as to limit the land use under a sustainable situation. To be consistent to the previous defined *AF* metric, we defined the condition for an *improved situation* by the logical AND of two conditions:

$$[AF(2050) - AF(t)]/AF(t) < 0.05 \quad AND \quad [AF(t) - AF(2014)]/AF(2014) < 0.20 \qquad (2)$$

In (2), *t* is the current year, i.e., when the decision on sustainability is made. We define the logical condition (2) the *constraint of land use sustainability (CLUS)*. *CLUS* is a Boolean variable and it is the results of the DANTHE decision-making support system. If condition (2) is true, *CLUS* is set to 1, otherwise it is set to 0. The constraint of land use sustainability has two parts: the first one expresses the relative change of *AF* between the current year and 2050 (*constraint for the future*), the second expresses the same relative change between the current year and the beginning of the time interval (2014). We call the second part *constraint towards the past*. The constraint for the future admits a maximum relative change of *AF* of 5%. This condition is used to preserve land by limiting the future impact of a single anthropic place on the environment. However, we can suppose that, in the horizon of 36 years, several building constructions will be delivered and that the system has to decide for more than one anthropic place. In order to avoid that several repeated limited building constructions on the territory (which would pass the constraint for the future) lead to a critical situation, the constraint towards the past admit e relative change of *AF*, if compared to the initial situation (2014), of 20%. In the second constraint, the relative change is higher (20% vs. 5%), because it includes all modifications made to the territory in the period [2014: *t*]. However, the limit of 20% allows the regions which start with a reversible degradation of land ($AF(2014) < 0.4$) to remain still to sustainable land use values. Obviously, if $t = 2014$, the second constraint will be always satisfied: the is no past, we are at the beginning of the time interval. Similarly, if $t = 2050$, there is no future in our model, and the only part of the constraint which survives is the second one. The terms *0.05* and *0.20* in (2) might seem too low: we have derived these values from several simulations of the system on the Italian territory, by comparing different areas and typology of new anthropic places. Besides, if the ecological consciousness of Italian and European citizens will rise in the future, and if the "no net land take" target of the European Union [18] will become a highly shared goal, probably these values should be even considerably reduced.

To move to a type C indicator, the ultimate challenge is to define the Response component in the DPSIR framework (Fig. 4), so that the indicator encompasses the largest possible number of states of the model. Let recall that the Response indicates some kind of action performed on driving forces, which is able to change the negative pressures on the environment. Since the DANTHE project is a decision-making support system, the action of Response is to guide the political organizations for territory planning control to make sustainable decisions on new human settlements (both rural or urban ones): a new human settlements, in a given region, will be built only if the *AF* indicators satisfies the constraint of sustainability of (2). Otherwise, the construction is not authorized by whom has the responsibility to preserve the territory environment. In this way, even if driving forces would push towards the new anthropic places, the response of the decision-making system suggests which of them are compatible to a sustainable land use for the territory, in 2050 near future, the last deadline for environmental preservation. Fig. 5 shows the role of the DANTHE system in the DPSIR framework for land use: the system considers nodes D, P, S and R. Node I is beyond the goals of this work, as the impact on health of humans and nature is very far from our research.

In order to use this new type C indicator, it is necessary to have a description of the new planned, but not already built, anthropic place. We denote by *description* a vector of data which describes the new anthropic place. Therefore, for a given region *S*, we define a new indicator, called *Dynamic Anthropentropy Factor* (*DAF*), as the following set of data:

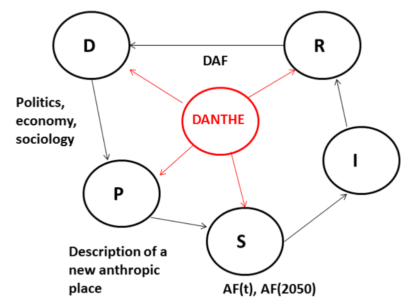


Figure 5. The role of the DANTHE system in the DPSIR framework for the problem of land use (D: driving forces, P: pressure, S: state, I: impact, R: response).

$$DAF = (t, AF(t), description, CLUS) \qquad (3)$$

where:

- *t* is the current year (*t* is a discrete variable in the range [2014:2050]) ;
- *AF*(*t*) is the *Anthropentropy Factor* (1) of region *S* at time *t*;
- *description* is a set of data which describes the new anthropic place which is going to be built in the region *S*;
- *CLUS* is the Boolean output (2) of the DANTHE decision-making support system: if $CLUS = 0$, the new anthropic place does not satisfy the constraint of land use sustainability. The condition $CLUS = 1$ means that the creation of the new anthropic place in region *S* is compatible with an innovative policy of environment preservation which takes into consideration a fair land use.

The DANTHE system performs the computation of the *DAF* indicator and generates the correct answer to the question: "will the new anthropic place be compatible to the environmental preservation target"? After the definition (3) of the new indicator *DAF*, it is possible to analyze how it is computed by the DANTHE system.

IV. The Architecture of the Decision-making Support System

In Fig. 6, the architecture of the DANTHE system is shown. The inputs of the system are the following:

- the visual map of the anthropic places, for the region of interest *S* under investigation (*Map1*). It is a Google Earth map, at the proper scale, where all the anthropic sub regions are delimited and colored in red. It represents the current land cover situation.

- The current year *t*, at which the input visual map *Map1* refer to; this time stamp is the starting point of the prediction and it is used in the condition (2).
- The demographic model used in the prediction (see Section V.A).
- The initial Anthropentropy Factor, i.e., *AF*(2014).
- The description of the new anthropic place: its position, shape and area are represented on the second visual *Map2*, at the same scale of *Map1*, while its type is represented by its class *i*.
- The class *i* of the new anthropic place, according to the classification of Table I.

Fig. 7a and Fig. 7b show examples of the input visual maps Map1 and Map2, respectively, for the island Vulcano (Eolie Islands, Sicily, Italy): in Fig. 7a, the red areas defines all the anthropic sub regions which are already present at time t, Fig. 7b shows the new anthropic place (in green), added to the existing situation, on the western coast.

The system architecture includes three modules: the computation, the predictive, and the decision module. The computation module computes the current Anthropentropy Factor AF(t), according to the method proposed in the ACI project [1] and summarized in Section II. The only difference, with respect to the previous computation, is that in the DANTHE project the calibration of the maps is no longer necessary, as the scale is derived directly from the Google Earth data.

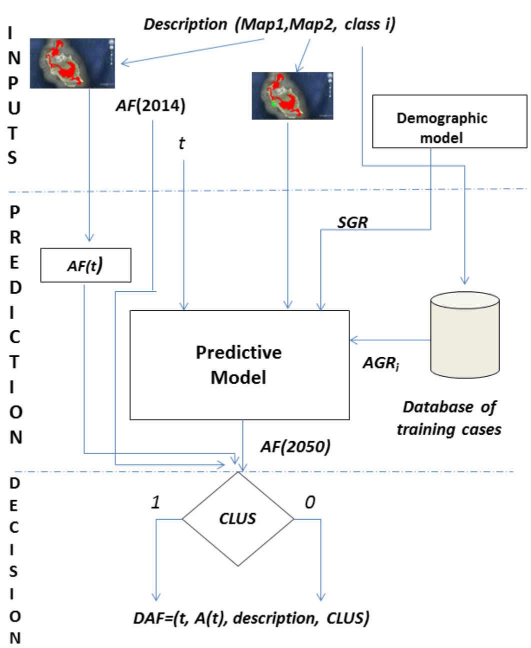
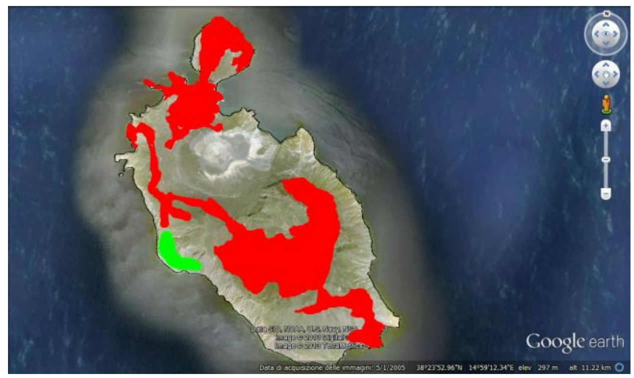


Figure 6. The architecture of the DANTHE system.

(a)



(b)

Figure 7. The input visual maps to the DANTHE system: (a) the visual map of the current anthropic sub regions (in red) of *S* (*Map1*). (b) the same map, with the addition of the new planned anthropic place (in green, *Map2*).

The second module is the predictive model, which predicts, starting from the input conditions, the value of *AF* at year 2050. The predictive model uses data derived by a database of “training cases” which present, for each class, a set of “worst cases” derived from the history of the Italian territory. The predictive model and its training will be discussed in detail in Section V.

The third module generates the output of the decision-making support system, i.e., the answer about the environmental sustainability of the new anthropic place. The module computes the logical condition of (2), i.e., the *constraint of land use sustainability (CLUS).*

It is clear that the intelligent core of the system is the predictive model and its assumption about the expansion of the *AF* for the different regions in the input maps.

V. THE PREDICTIVE MODEL

The predictive model computes the value of *AF* for the year 2050; actually, it can generate a predicted value of *AF* for every year, from the current year *t* to 2050, but the output refers only to 2050, as it is used by the decision module to compute the CLUS value. In order to compute *AF*(2050), the model has to depict the situation of the anthropic places, i.e., it has to foresee how they will grow in time. The prediction has to answer two different questions:

- How will expand the area of the *existing* anthropic places?
- How will expand the area of the *new* anthropic place?

For example, if we consider the case depicted in Fig. 7b, the predictive model has to compute the future situation for the "red" regions and for the "green" region. After the prediction, the system applies standard dilation of 50 meters, in order to compute the future Dead Zone and, consequently the predicted *AF* value, according to (1). The predictions of the area expansion, both for the existing and for the new anthropic places, are based on different assumptions.

A. Assumptions on the expansion of existing areas

For what concern the existing anthropic places, they are described by their shape and dimensions, without any assumption of what they are. For example, in Fig 7a, we know only that the red regions represent anthropic places, but we do not know where there are houses, industries, touristic settlements or commercial centers. Obviously, red regions contains hundreds, or more, anthropic places belonging to every class of Table I. The definition of "existing anthropic places" does not contain information about the type, because the input of the system (*Map1*) is only a visual map with red regions. For this reason, it is not possible to make assumptions on the growth of the areas by considering some elements related to the type of anthropic places, as classified in Table I. The only information we have are the regions, their shapes and their relative positions. For this reason, we can make the following most general assumption: the existing areas grow, without a preferred direction, according to a growth rate. Therefore, on red regions, the model performs a standard morphological operator [12], a circular dilation, to enlarge the total red areas according to a specified growth rate. What is a sensible value for this parameter? We cannot use average annual land take indexes, as measured by several sources [3-5]. In fact, these values are the result of a dramatic land take on the territory, in the last years. As we are interested in sustainability, our model has to make an assumption which is coherent with the goal of a controlled land take. Land is a precious resource, and the only reason a *green* society has to consume this resource is a demographic expansion. Without a positive growth of populations, existing anthropic places should not expand. This is far off of being a radical position: it is well known in the field of urban planning and urbanization [19], as supported also by famous architects, such as Renzo Piano [20], who syntheses this concept in an meaningful sentence: "*stopping the expansion of the city by explosion and starting implosion. Growth of the city from inside*" [21]. By adopting this rationale, the DANTHE project uses a projections estimate of the resident population. There are several projection models on population growths [22-24], but the majority estimate, for Italy, a demographic expansion until 2015; later, a negative population growth until 2050. For example, a detailed projection from ISTAT [24], the Italian National Institute of Statistic, shows only an increment for years 2014 and 2015, equal to 0.11% a year, followed by a decrement of population. By transferring this rate on the areas, we can foresee an increment, for the entire horizon [2014-2050], of 0.22%. The DANTHE project uses these data, thus assuming that the existing anthropic places areas have to expand according to a *Sustainable Growth Rate* (*SGR*) of 0.22%. Obviously, this value is parametric, and the predictive model can accept in input other demographic projection models, and, consequently, other values for *SGR*.

B. Assumptions of the growth of the new area

For the new anthropic place, the model can do better that a computation based on an parametric growing rate, because it can exploit the important information about what *type* of anthropic place is: a new set of cottages, a stadium or a large commercial center? The model considers peculiar characteristics of the new anthropic place to perform a customized prediction, based on some intrinsic features: the *initial area*, the *type* and the *relationship* to existing surrounding settlements.

The *initial area* of the new anthropic place is the green area depicted in *Map2* (see Fig. 7b): it determines the initial impact on the environment. Unfortunately, in most of the environmental compatibility studies, this is the only element. This means that usually the evolution in time is completely ignored. Instead, the DANTHE model cannot limits its prediction to a static evaluation of the situation, as it is based on a dynamic indicator. For this reason, the system considers other elements to predict the enlargement of the new area. First of all, the *type* of the anthropic place: a large commercial center will grow at a fast rate, because of the addition of new shops, parking, restaurants, and other services. The ability to attract economical investments will transform this type of anthropic place itself as a magnet for new settlements. A reasonable assumption is that the growth will be greater for a large commercial center than for other types of settlements, such as small shops, schools, or cemeteries. We call the property of attracting other new anthropic places the *anthropogenic characteristic*: some types of new anthropic places are able to *generate*, in a sufficient number of years, a consistent numbers of other anthropic places (this is the reason of the suffix *genic* in the *anthropogenic* term). According to this assumption, the DANTHE model classifies the new anthropic places in twelve classes (Table I): inside a class, different human settlements can be present, but all of them share the same anthropogenic characteristic. For example, in class 1, a school is something very different from a cemetery, but both have the same *low growing* behavior. The terms *low growing* and *fast growing* (Table I) for the similar typology of human settlements differentiate their anthropogenic characteristic. The classification of Table I has been confirmed by the study of the history of the Italian territory. The model is able to compute the future configuration (shape and area) of the new anthropic place only if there is an estimation about the annual growing rate of land use, expressed as the relative increment of area for each year of prediction. According to the basic assumption that the Annual Growing Rate is specific of the class *i* the new anthropic place belongs to, let denote this parameter of the model as the *Annual Growing Rate*, or AGR_i, for the class *i*: its numerical values will be estimated in the training phase, as explained in Section VI.

The initial area and the class (and, consequently, its Annual Growing Rate) of the new anthropic place determine the area it will occupy at the prediction time. However, they do not determine the shape of the expansion. As *AF* computation is based on morphological operators, shape is fundamental to determine the final result. The third characteristic of the expansion of new anthropic places considers this element: how far the surrounding area influences the future shape of the new human settlements? The model gives three possible answers: no influence, gravity influence and road influence.

No influence means that the new anthropic place is not influenced by the surrounding existing anthropic places, due to the distance or because, for its intrinsic typology, the new anthropic place tends to enlarge its shape without a preferred direction and independently from the existing situation. In this case, the model applied to the initial area (green area of *Map2*) a classical operator of mathematical morphology: the dilation with a circular structural element. It is the same operator used in the computation of the static *AF*, to determine the Death Zone. Fig. 8 shows an example of *no influence growth*. The green area is the new anthropic place, the red area is an existing anthropic place; the new anthropic place enlarges its shape with a circular symmetry, regardless the existing "red object". After a certain period, the new anthropic place will have the same initial shape, with a symmetrical enlargement in every direction (0-360 degree).

Gravity influence means that the new anthropic place is influenced, in its growth, by the surrounding existing places, because it is attracted by the mass center of existing area, as two objects are attracted by the universal gravity law. For example, small shops and human settlements (class 1) tend to be attracted to the surrounding urban settlements. Fig. 9 shows an example of gravity influence: the new place grows in the direction of the line which ideally conjugates the two mass centers. This type of growth is implemented by a modification of the standard dilation with a circular structural element: the algorithm computes the two mass centers, the equation of the line and performed a reduced circular dilation, where only a subset of directions are considered; the directions change as the dilated area gets closer to the existing area, as the gravity force increases. We call this new morphological operator *gravitational dilation* (Fig. 9).

Figure 8. The result of a dilation with a circular structural element. The original area (in green) is enlarged in each direction with a circular symmetry. In red, the existing anthropic place, which does not influence the new area (in light blue).

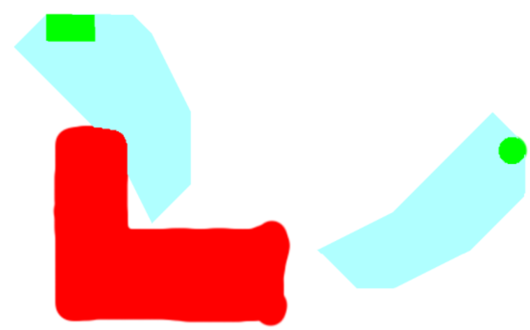


Figure 9. The result of the new morphological operator *gravitational dilation*. The original areas (in green) are enlarged in the direction of the existing anthropic place (in red) as if they were attracted by its mass center.

Road influence: roads are very important to define the future development of anthropic places. They are a catalyst for the expansion of urban and rural settlements, as it is much more probable that new settlements will develop along existing roads to facilitate travel and communications. This is particular true for certain type of anthropic places, such as houses and residential buildings. The model defines and implements a *road dilation*, where the new area grows in parallel to the road (see Fig. 10). In the current implementation of the DANTHE model, the road dilation can considers up to three main roads, surrounding the new object.

By analyzing several real cases of the Italian territory, we can make the assumptions that circular dilation is typical of anthropic places of class 4, 6, 7, and 8, gravity dilation is typical for class 3, 5, 9, 10, 11 and 12, and road dilation is typical for class 1 and 2.

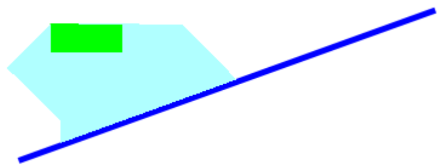


Figure 10. The result of the modify morphological operator *road dilation*. The original area (in green) is enlarged along of the road (in dark blue).

C. The predictive algorithm

The predictive algorithm computes the new area both for the existing and for the new anthropic places at time $t + \Delta$, starting from the initial conditions at time t ($t >= 2014$, $t+ \Delta <= 2050$).

For the existing anthropic places (*Map1*), the algorithm performs a dilation with a circular structural element, until the new area is incremented by a factor equal to the *Sustainable Growth Rate* (*SGR*) , which is provided by the adopted demographic model [24].

For a new anthropic place (green part of *Map2*), belonging to class i ($1 <= i <= 12$) of Table I, the future area at time $t + \Delta$ is computed by the following algorithm:

- *STEP 1:* For each pixel of the initial region, perform a circular dilation (if i = 4, 6, 7, 8), a gravitational dilation (if i = 3, 5, 9, 10, 11,12) or a road dilation (if i = 1, 2).
- Compute the area of the new dilated region.
- Go to *STEP 1*, until the new dilated area grows of a factor $AGR_i *\Delta$

So far, in order to implement the algorithm, the system needs a set of values for AGR_i. They have been estimated by analyzing the history of the Italian territory (see Section VI).

VI. Experimental results

We describe two kinds of experimental results: (a) data analysis for the estimation of the Annual Growth Rate for classes 2, 3, 4, 6, and 7, and (b) a real case of application of the decision-making support system. In order to test the basic assumptions of the model and the three types of dilation (circular, gravitational and road), we have analyzed significant cases of the history of the Italian territory. The events occurred in the past are the basis for the estimation of the *Annual Growth Rate*. This is called training phase, because the model has to learn, from the past, reasonable values for the future predictions. The phase involves an analysis of Italian territory of many examples of new settlements that have been built. For each case, we have analyzed the situation at a certain time t' and, for the same area, at t'', using the built in Google Earth function *View > Historical Imagery*. In this case, the analysis is related to the past, so both t'' and t' are less than 2014. The values of t' and t'' may vary for the cases, as we need to study situations where Google maps are available and where the anthropic place has developed its maximum growth (otherwise, it would not be a worst case). By comparing the area of the settlement at time t'' and time t', it is possible to estimate the *Annual Growth Rate*. Let define $A(t')$ and $A(t'')$ the two areas for a given anthropic place of class **i**. We define the Annual Growth Rate for class i as:

$$AGR_i = [A(t'') - A(t')] / [A(t')*(t''- t')] \qquad (4)$$

Obviously, each case of the same class, provides a different value for AGR_i. One possibility is to choose a sufficient number of cases in the Italian territory and compute average values. However, this method is very time consuming and would make the estimations of AGR_i vulnerable to a number of probabilistic fluctuations due to plenty of factors: the great variety of the Italian territory, environmental, social and economic factors. Overestimating or underestimating AGR_i would lead to unreliable predictions of the model. For this reason, we do not follow a statistical approach (choice of a sufficient number of cases and averaging the computed values of AGR_i). Instead, we perform a "worst case analysis". We choose, for each class, a case which is, for some peculiar characteristic, the "worst case" we can find: in this way, we know that the prediction will be severe, but not unrealistic, because, unfortunately, similar cases have already occurred in the past. The worst cases allow to store in the model values of AGR_i for all the classes. The training worst cases can be updated, if in the future we will able to find even worst case. The database of our model contains all the worst cases (visual maps and data) and is, actually, a description of some of the most terrible insult to the environment, from a land use preservation point of view.

A. The training phase: significant cases

The first training worst case is for class 2 (see Table I), i.e., fast-growing settlements, such as houses, villas, cottages, mansions, including small parking lots. We have chosen the small town of Cura Carpignano (Pavia, Lombardia). Located in east land around Pavia, near the river Olona, it is a satellite town of the bigger city of Pavia. Cura Carpignano had a large population growth [25] during the last decade (Fig. 11). This increase has led to double its population in the time interval [2001:2011], with a consequent growth of the local construction industry.

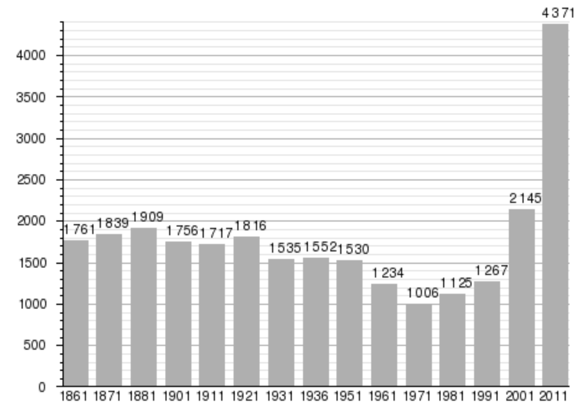


Figure 11. The population growth of Cura Carpignano (Pavia, Lombiardia). The boom is in the years 2001-2011.

What makes Cura Carpignano a worst case? The architectural choices led the town to expand in a purely horizontal direction (no buildings over three floors); as a consequence, most of the rural and agricultural lands around the village have become residential zoning, with a dramatic land take. Besides, the small area of the town and its location allows us to make precise measurements, without any influence due to nearby settlements. We analyzed the Google Earth maps history for t'= 2003 and t'' = 2012 and we compute the *AGR* according to (4). As this type of anthropic place belongs to class 2, the resulting value is AGR_2. In nine years, from 2003 to 2012, the area has grown from 0.29 square kilometers to 0.43 square kilometers, with an increasing rate of 48.2%. This is much greater than the value of rural land consumption of 8.8% for the same decade from [26]. The results confirm that this is really a worst case. In Fig. 12, a plot of the area of Cura Carpignano is shown: in red we have covered the anthropic area in 2003, in dark blue there is the new area subtracted to nature in nine years. This training case provides a value for AGR_2 = 0.053639.

Figure 12. The growth of the residential settlement in the territory of Cura Carpignano, Pavia, Lombardia, Italy (in red: initial area in 2003, in blue the increment of area in 2012).

Figure 13. The area of the commercial center Euroma 2, Rome (in red: initial area in 2009, in blue the increment of area in 2013).

The second worst case refers to class 3, Commercial centers. We chose *Euroma 2*, a commercial complex located near Rome; as it is considered the largest commercial center of Rome (and in Europe, when it began to be built), it represents a worst case [27]. In only four years (2009-2013), the area of land use increased by 58% (see Fig. 13). This was due to the fact that the center has attracted the construction of some stores, a residential complex, a tower and a sports center. This training case provides a value for AGR_3 = 0.145.

The third worst case refers to class 4, Industrial areas (Table I). We chose the industrial center near Osoppo (Udine, Friuli-Venezia Giulia), an Italian town of 3,016 inhabitants. We chose the case of Osoppo because, in recent years, it became the focus of many environmental battles and economic discussions because of its huge industrial center, which currently occupies an area of 2,316,125 square meters. We analyzed the growth of the industrial center, from 2002 to 2012: we obtained a growth of 38.4%, from 1.3 square kilometers to 1.8 square kilometers. In Fig. 14, a plot of the area of Osoppo industrial center is shown: in red we have covered the anthropic area in 2002, in dark blue there is the new area subtracted to nature in ten years. This training case provides a value for AGR_4 = 0.03846.

The fourth worst case refers to class 6, fast growing service area (Table I): we applied our study methodology to the case of the campus of the University of Pavia, in the north west part of the city. It was built in the mid-80s and expanded several times. Its peripheral location, if referred to the old city, allows us to give a correct estimate of such expansion, without influences or constraints on the construction details. In eleven years, there has been an area increase of 95.3%, from 64,000 to 125,000 square meters. During these years, there have been several constructions and expansions: the parking areas in the south, a new museum, a research center, but also a swimming pool and several residential settlements. This case really shows the meaning of the term *anthropogenic place*: the initial situation of 2002 (Fig. 15) attracts new anthropic places (Fig. 16), with a land take of 95.3% in eleven years.

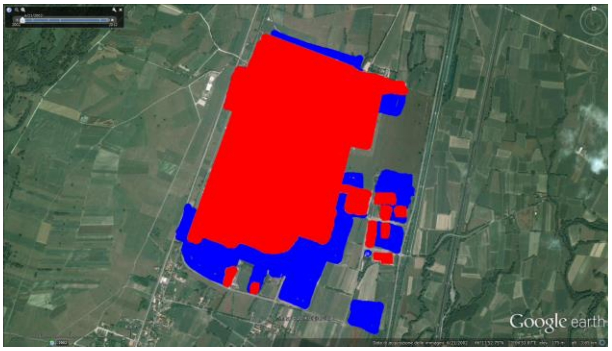


Figure 14. The area of the industrial center of Osoppo (in red: initial area in 2002, in blue the increment of area in 2012).

Figure 15. The area of campus of the University of Pavia, in the north west periphery of the city, in 2001.

Figure 16. The area of campus of the University of Pavia, in the north west periphery of the city, in 2012.

Figure 17. The area of the campus of the University of Pavia (in red: initial area in 2001, in blue the increment of area in 2012).

In Fig. 17, a plot of the area of the campus of University of Pavia is shown: in red, we have covered the anthropic area in 2001, in dark blue, there is the new area subtracted to nature in eleven years. This training case provides a value for AGR_6 = 0.086647.

The last worst case here described is related to class 7, Airports and heliports. We chose the Galileo Galilei Airport (Pisa, Tuscany), the main airport of Tuscany for number of passengers, the second in Central Italy after Rome-Fiumicino. Initially used only for military purposes, the Galilei Airport had a significant expansion during the 90-s, following the opening of low cost airlines.

In particular, from 2002 onwards, satellite maps allow us to estimate the strong expansion of the airport, runway and external warehouses, which have completely cemented the western area. In ten years it has gone from an area of 1,209 km square kilometers to an area of 2,608 square kilometers, with a percentage increase of 115.7%. This training case provides a value for AGR_7= 0.115715.

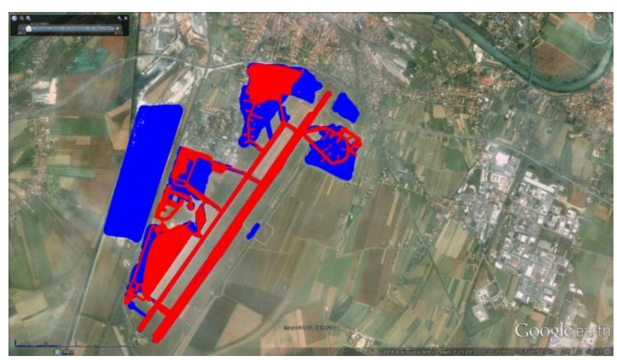


Figure 18. The area of Galileo Galilei Airport, Pisa, Tuscany (in red: initial area in 2002, in blue the increment of area in 2012).

B. An example of prediction

After the training phase, it is possible to see the result of the DANTHE project on a challenging case. Let suppose that in Cura Carpignano a new residential settlement is proposed: this is the same town we have used for a the first training worst case (Section VI.A). As we have seen from the past urban planning, in Cura Carpignano there is a great demographic expansion (Fig. 11). This trend is still in place, as the most recent data (2013) shows that the population is increasing, with respect to data of Fig. 11 (population at 2013: 4590), even if with a less growth rate. Therefore, the hypothesis of a new residential settlement is realistic. We processed the Google Earth map, at a proper scale, to generate the images which describe the present and future, planned situation. Fig. 19 shows the red areas, corresponding to the present situation (t = 2014) with the existing anthropic places: it corresponds to the first visual input *Map1*. Fig. 20 shows where the new settlement is expected (green area); it corresponds to the second visual input *Map2*. Also the position of the new settlement is realistic: at the periphery of the main urban area and close to roads.

The DANTHE system answers the question: "is the settlement sustainable"? The system performs the circular dilation according to the previous definition of Sustainable Growth Rate (*SGR*) on the red areas, and a road dilation following the two main roads close to the new settlement. The result (Fig. 21) represents the prediction of the new area enlargement at year 2050. The two roads used in the road dilation are shown in dark blue. Fig. 22 shows all the territory of the municipality Cura Carpignano, as predicted according to the DANTHE model: both the two expansions are combined. After the prediction, the system computes the two values of Antropentropy Factors which are necessary to obtain *CLUS* indicator (3). The system performs the circular dilation (radius of 50 meters) on *Map1* for the definition of the corresponding Death Zone, and the computation of *AF(t)* according to (2). Subsequently, the system performs the same data processing on the result of prediction (Fig. 22) to compute *AF(2050)*. The corresponding Death Zone of the predicted total area in 2050 is shown in Fig. 23. The numeric results are 0.25 and 0.271 for *AF(2014)* and *AF(2050)*, respectively. By substituting these values in (2), we obtain for the condition

$$0.084 < 0.05 \ \ AND \ 0 < 0.2. \qquad (5)$$

As condition in (5) is false, *CLUS* = 0 and the settlement has been rejected by the DANTHE system. According to the previous definitions, at the end of the prediction the DANTHE project outputs the new indicator, i.e., the *Dynamic Anthropentropy Factor*, *DAF* (3) :

$$DAF = (2014, 0.25, description, 0). \qquad (6)$$

In (6), *description* is the set of data (*Map1*, *Map*, *2*); moreover, the new indicator *Dynamic Anthropentropy Factor* means that this new anthropic place, of class 2 (Table I, "Fast-growing settlements"), planned in 2014 in a municipality of current *AF* = 0.25, *is rejected*, because it is not compatible to the assumptions of environmental sustainability (the last component of (6), i.e., *CLUS* is 0).

Figure 19. The visual input *Map1* to the DANTHE system for the prediction on a new residential settlement in Cura Carpignano: the existing situation at t=2014.

Figure 20. The visual input *Map2* to the DANTHE system for the prediction on a new residential settlement in Cura Carpignano (in green: the new planned anthropic place).

Figure 21. The result of the prediction on the new settlemet (in red: the existing anthropic places, in green the original new planned anthropic place, in light blue the increment according to road dilation, in dark blue the road directions).

Figure 22. The total antrhopic places for the whole territory of municipality of Cura Carpignano, acoording to the prediction for year 2050.

Figure 23. The Death Zone corresponding to the total anthropic places of Fig. 22 (in white, *AF(2050)* = 0.271).

C. Discussion about the quality of the prediction

How is it possible to check the quality of the prediction of the DANTHE system? As remarked by the European Environment Agency [6], the target for environmental sustainability is the year 2050, and our system adopts this choice. Besides, it is intrinsic in the historical definition of sustainable development (*"Development that meets the needs of the present, without compromising the ability of future generations to meet their own needs"* [28]) that, in order to define a decision as a "sustainable one", we have to choose a medium or long term temporal period of study. Obviously, the check of the quality of the prediction cannot be a *direct* check, because we cannot wait until 2050 to verify if the prediction is true, and in which extent! Moreover, even the application of the decision-making support system in a *retroactive way* would not be correct, because it would be inconsistent with the assumptions. In fact, if we applied the prediction to a generic situation of the past, and if we compared the prediction to the present and real, situation, this should not be enough to prove the validity of the prediction for the future, because the starting conditions (i.e., population growth, present land use) should be different. For all these reasons, we can estimate the quality of the prediction upon a probabilistic approach, by making assumptions and proposing recommendations that make the conclusions of the prediction system reasonable and convincing. The first goal has been reached by a careful choice of the worst cases in the training phase. The second aspect involves the use protocol of the system. It is based on a double-check of quality and a certification mechanism, and it is strictly related to the new choice of distributed client-server architecture, as explained in Section VI.D.

D. Web platform and use protocol of the DANTHE system

One of the innovation of the DANTHE project, if compared to the previous ACI project [1], is the software platform used to implement the computation of *AF* and *DAF* indicators. In the previous approach, we used a *User Generated Content* (*UGC*) and a social crowdsourcing paradigm: we involved a social networking community in the project, in order to generate open data. Users of the social network were asked to use Google Earth Maps to generate the maps of anthropic places (according to the previous definitions, the *Map1* image). However, in order to compute *AF* (2), these maps were collected in a centralized point (the University of Pavia) in order to use Matlab code to implement the morphological operators. By collecting the input maps to the centralized point, a careful check of calibration of the map scale and of consistency of input data were performed, before the computation of *AF* (2).

In the DANTHE project, we still use the *UGC* paradigm, but the computation of *AF* and *DAF* indicators is distributed. The algorithm for standard and modified morphological operators (circular, gravitational and road dilations) and for the implementation of all the parts of the DANTHE architecture (Fig. 6) have been written on an open source platform (java and php) in a Web-based system. The goal is to allow a user to connect to a server and use only the browser in order to submit data and compute both *AF* and *DAF* indicators. The *UGC* paradigm is still valid, as the user is asked to create and submit to the system the two maps (*Map1* and *Map2*) and the class of the new anthropic place. The check of consistency of the scale of the map is performed automatically by the software, as the scale information are derived directly by Google Earth scale indicator. Moreover, the other quality checks for the input maps (*Map1* and *Map2*) are performed by validating the users, instead of the data, by using a *certification* mechanism. The central operating unit of the University of Pavia certificates and authorizes the user of the DANTHE system after the check that: a) the user has followed a training course on the correct use of the system, b) the user is able to generate input maps which are consistent to the assumptions of the system and strictly adhere to a specified level of detail and visual quality and (c) the user is able to exploit the decision-making support system in a step-by step protocol of correct use. We define this quality check a *double-level* *certification*, because it addresses *two* classes of users: *UGC creators*, who are responsible for generating the maps of *AF* computation for the reference year *AF(2014)* and for the current year (if different, *AF(t)*), and the *super users*, who are interested in using the decision making support system, to decide if a future anthropic settlement will be sustainable. The check of quality is double: on the *UGC creators*, which generate contents for the static computation of *AF*, and on the *super users*, who actually are interested in the DANTHE system capability of decision support about sustainability. Who are the potential users of the DANTHE systems? *UGC creators* and *super users* do not necessarily belong to the same set: creators can be scattered all over the national territory and they are responsible for carefully reporting the actual conditions and variations of the land use in a given municipality. In this way, the system can react rapidly to local changes. Examples of *super users* are the subjects who are involved and can influence the decisions in urban planning. By referring to the DPSIR framework (Fig. 5), the *super users* can be all the subjects in the society who have the power to express the right *Response*, in order to react to the *Driving Forces* which request unsustainable land use: for example, organizations and institutions, local and central political entities, and government environmental agencies.

On the time of writing the final version of the present paper, the DANTHE system has been completely developed, and also the Web visual interface (in Italian) has been completed (Fig. 24). Current work is (a) the formalization of the certification process to activate the trusted *UGC creators* and *super users*, and (b) the definition of a database that collects all the cases examined by the DANTHE system during the certification phase. At the end of these two last steps, the system will be delivered over the Web for intensive use by the (trusted and certificated) *UGC creators* and *super users*.

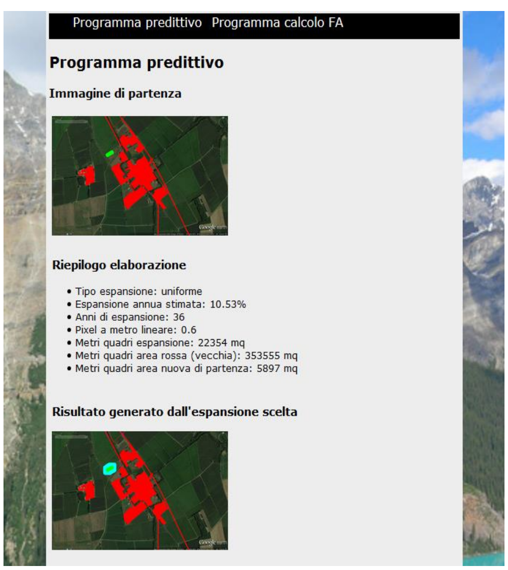


Figure 24. The Web interface of the DANTHE system.

E. Using the DANTHE system for recommended positions

Beside the prediction of sustainability of future anthropic places, presented in Section VI.B, the DANTHE system can help in deciding the *right* position of a new anthropic place. This is a sort of support for a *recommended* position: as pointed out in Section V.B, the *DAF* indicator considers not only the initial area of the new anthropic place, its anthropogenic characteristic and shape, but also the surrounding situation of existing anthropic places. Therefore, it is possible, by simply moving the position and shape of a new settlement, to transform a verdict of unsustainability (*CLUS* = 0) to a positive verdict of sustainability (*CLUS* = 1). Let consider the case depicted in Fig. 25: a new settlement (in green) is positioned in the north west surrounding area, outside the existing urbanization (full red areas). This starting point leads to a predicted situation in 2015, which the model considers unsustainable. In fact, the system computes for (2) the following condition:

$$0.0681 < 0.05 \text{ AND } 0 < 0.2. \tag{7}$$

As condition in (7) is false, *CLUS* = 0 and the settlement has been rejected by the DANTHE system. However, if we choose another starting condition, with an equivalent new anthropic place, with the same dimension and anthropogenic class, but different shape and position, the prediction can be reversed: if the new anthropic place (see Fig. 26) fills the "hole", the expansion in 2050 will preserve land use, as it will occupy an area which is incorporated inside the expansion of the existing areas. In this case, the system computes for (2) the following condition:

$$0.0147 < 0.05 \text{ AND } 0 < 0.2. \tag{8}$$

As condition in (8) is true, *CLUS* = 1 and the settlement has been approved by the DANTHE system. The final situations in 2050 are compared in Fig. 27: by filling the hole (on the right), the predicted Death Zone is reduced, if compared to the unsustainable situation (on the left).

Figure 25. An example of unsustainable settlement (in green: the original new planned anthropic place, in light blue the dilated area).

Figure 26. The same new anthropic place of Fig. 25, with a different position and shape, but equal dimensions: the growth "fills" the hole of fragmented original areas.

Figure 27. By moving the position of the settlement, it becomes sustainable: the corresponding Death Zone in 2050, for starting condition of Fig. 25 (on the left) and Fig. 26 (on the right), respectively.

VII. CONCLUSION AND FUTURE WORK

In this paper, an innovative indicator to evaluate land use is proposed, based upon the new concept of *Dynamic Anthropentropy*. It is an improvement of the previous indicator, the *Anthropentropy Factor*, defined in the ACI project. It is used in the proposed decision-making support system (DANTHE), which allows to discover if a new building construction will be compatible, in its dynamic expansion, to the target of environmental sustainability of a controlled land use. This target is one of the most challenging aspect of the policies of European Union and of the European Environmental Agency, for the time deadline of the year 2050.

The system performs a prediction of land use and compute the new indicator to make the decision of sustainability. It has been trained with cases, taken from the history of the Italian territory, in order to estimate important parameters of the prediction. The system uses also a demographic model, as the projection of future population is related to another important parameter: a sustainable growth rate of existing anthrophized areas.

Experimental results have shown the predictions and the evaluations of the consequences of new urban expansions in the territory, not only for what concerns their initial impact, but also for the temporal evolution, until year 2050.

Current work is the refinement of the certification process to create a community of trusted users and super users, in order to assure the quality of prediction by validating the cultural background and skill of use of the system, in order to generate affordable predictions.

Future work will be the complete delivery of the system over the Web, and its application to other critical cases of urbanization planning of Italian territory, to further prove the usability and the performance of the system.

ACKNOWLEDGMENT

The authors greatly appreciate for his collaboration Antonio di Gennaro, for map generation from Corine data set, and Gabriele Porro, for the Web implementation of DANTHE model in php technology.

REFERENCES

[1] Maria Grazia Albanesi and Roberto Albanesi, "A new approach based on computer vision and collaborative social networking for environmental preservation: theory, tools and results of Italian ACI project," Proceedings of The Eighth International Multi-Conference on Computing in the Global Information Technology ICCGI2013, Nice (France), 21-26 July 2013, pp. 16-21, Copyright (©) IARIA, 2013, ISBN 978-1-61208-283-7.

[2] European Environment Agency, "Urban sprawl in Europe - The ignored challenge," EEA Report n. 10/2006, http://www.eea.europa.eu/publications/eea_report_2006_10 [retrieved: May, 2014].

[3] Overview of exiting policies to reduce and mitigate soil sealing in the EU and Member States, http://ec.europa.eu/environment/soil/pdf/sealing/4.%20Overview%20of%20existing%20Policies%20EU%20and%20MS%20(A-H).pdf [retrieved: May, 2014].

[4] Land use, state and impact (Italy), http://www.eea.europa.eu/soer/countries/it/land-use-state-and-impacts-italy [retrieved: May, 2014].

[5] ISPRA, "Urban environment quality, Report IX," Transl. "Qualità dell'ambiente urbano, IX rapporto," 2013, http://www.isprambiente.gov.it/it/pubblicazioni/pubblicazioni-del-sistema-agenziale/qualita-dellambiente-urbano-ix-rapporto.-edizione-2013 [retrieved: May, 2014].

[6] European Environment Agency, "Towards a green economy in Europe - EU environmental policy targets and objectives 2010–2050," EEA Report n. 8/2013, http://www.eea.europa.eu/publications/towards-a-green-economy-in-europe [retrieved: May, 2014].

[7] J. Dumanski and C. Pieri, "Land quality indicators: research plan," Agriculture, Ecosystems & Environment 81.2, 2000, pp. 93-102.

[8] K. Wiebe, Land quality, agricultural productivity and food security, 2003, Edgar Publication.

[9] J. D. Majer, "Ants: bio-indicators of minesite rehabilitation, land-use, and land conservation," Environmental Management, Vol. 7, n. 4, Springer Verlag, 1983, pp. 375-383.

[10] H.G. Zechmeistera, I. Schmitzbergera, B. Steurerb, J. Peterseila, and T. Wrbkaa, "The influence of land-use practices and economics on plant species richness in

meadows," Biological Conservation, n. 114, Elsevier, 2003, pp. 165–177.

[11] The UN convention on biological diversity, http://enrin.grida.no/biodiv/biodiv/cbd/eeafoll.pdf [retrieved: May, 2014]

[12] R. C. Gonzales and R. E. Woods, Digital image processing, Pearson Prentice Hall, 2008, Chapter 9, "Morphological image processing".

[13] Corine Land Cover Project, http://www.eea.europa.eu/publications/COR0-landcover [retrieved: May, 2014]

[14] J. Rogan and D.M.Chen, "Remote sensing technology for mapping and monitoring land-cover and land-use change," Progress in Planning, n. 61, Elsevier, 2004, pp. 301–325.

[15] A. Veldkamp and E.F. Lambin, "Predicting land-use change," Agriculture, Ecosystems and Environment, n. 85, Eslevier, 2001, pp. 1–6.

[16] L. Fang, Q. Zhuping, A. Zengxiang and W. Xianhu, "Urban expansion and evolution prediction of Jinan city based on remote sensing and GIG technology," Proc. Of Urban remote seninsing joint event, IEEE Press, 2009, pp. 1-6, doi: 10.1109/URS.2009.5137633.

[17] E. Smeets and R. Weterings, "Environmental indicators: typology and overview," European Environment Agency Technical Report n. 25/1999 http://www.eea.europa.eu/publications/TEC25 [retrieved: May, 2014]

[18] Communication from the commission to the European Parliament, "Roadmap to a resource efficient Europe," http://ec.europa.eu/environment/resource_efficiency/pdf/com2011_571.pdf [retrieved: May, 2014]

[19] S. Lehmann, "Sustainability on the urban scale: "green Urbanism"," International Conference on Sustainable Architectural Design and Urban Planning Hanoi Architectural University, May 15-16, 2007, Hanoi, Vietnam, http://unaus.eu/pdf/U010.pdf [retrieved: May, 2014].

[20] Renzo Piano curriculum vitae, http://www.fondazionerenzopiano.org/a/46/renzo-piano/?l=en [retrieved: May, 2014].

[21] Interview: Renzo Piano on The Shard, http://www.dezeen.com/2012/05/18/interview-renzo-piano-on-the-shard/ [retrieved: May, 2014].

[22] United Nations, Department of Economics and Social Affairs, "Probabilistic population projections," http://esa.un.org/unpd/ppp/Figures-Output/Population/PPP_Total-Population.htm [retrieved: May, 2014].

[23] ISTAT, "Population projections estimate of the resident population," http://www.demo.istat.it/prev/index2_e.html [retrieved: May, 2014].

[24] ISTAT, "National population projections," Transl. Previsioni demografiche nazionali, http://www3.istat.it/salastampa/comunicati/non_calendario/20060322_00/nota%20previsioni.pdf [retrieved: May, 2014]

[25] http://www.comuni-italiani.it/018/060/statistiche/popolazione.html [retrieved: May, 2014].

[26] ISTAT, Agricultural land and land consumption, 2013, http://www.istat.it/it/archivio/107569 [retrieved: May, 2014].

[27] http://it.wikipedia.org/wiki/Euroma_2 [retrieved: May, 2014].

[28] Report of the World Commission on Environment and Development, Our Common Future, http://conspect.nl/pdf/Our_Common_Future-Brundtland_Report_1987.pdf [retrieved: May 2014]

International Journal on Advances in Intelligent Systems, vol 7 no 1 & 2, year 2014, http://www.iariajournals.org/intelligent_systems/

Does the Administrator Community of Polish Wikipedia Shut out New Candidates Because of the Acquaintance Relation?

Justyna Spychała
Faculty of Polish Studies
University of Warsaw
Warsaw, Poland
Email: jspychala@gmail.com

Mateusz Adamczyk
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Warsaw, Poland
Email: adamczyk@mimuw.edu.pl

Piotr Turek
independent
Warsaw, Poland
Email: piotrt@gmail.com

Abstract—**Administrators of Wikipedia are its most dedicated users, which are granted special privileges and burdened with great responsibility for the Wikipedia. The administrators are usually nominated by the community and then elected by voting or by reaching a consensus. This paper examines period 2005 – 2011. In most recent years of examined period, decline in a number of newly appointed administrators can be observed. There are two main hypotheses of seen phenomena: with growth of Wikipedia it is harder to become an administrator or existing administrator community is shutting out new candidates due to acquaintance relation. This research is an attempt to find out whether the community is not becoming less open to new users and new potential administrators, because of their lack of chances to gain reputation. The key here is the understanding of social aspects driving the process of Request for Adminship (RfA) votings. Based on our previous work focused on social networks induced from collective activity of wikipedians, this paper extends it by the annual analysis of obtained statistics and examination of clustering coefficient as an approximation of social capital. We present the dynamics of relationships between voters and candidates across several years of Polish Wikipedia development. Obtained data allowed us to answer the question, whether administrator community of Polish Wikipedia shuts out new candidates because of the acquaintance relation.**

Keywords—Multidimensional Behavioral Social Network; Wikipedia; Request for Adminship; Clustering Coefficient; Administrators' Lifespans; Administrator Community versus core; Wikipedia's users behavioural patterns.

I. Introduction

Spychała et al. [1] presented results of their research on The Third International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA 2013 [2]) in Nice, France. Presented research concerned possibility of closing up the Administrator Community of Polish Wikipedia by shutting out new candidates. It was a refinement of work done by Turek et al. [3]. This paper is an extended version of paper written by Spychała et al. [1] and presented on COLLA 2013. Similarly to paper written by Spychała et al. [1], this paper covers the range of years 2005-2011 and its goal is to answer the question if the reason for decrease in successful RfA votings in Polish Wikipedia is choosing Administrators based on acquaintance. As the administrator community of Polish Wikipedia is rather small, especially compared to the English one, it could be valid claim. But, we argue that it is not the case. Probably, it is caused by growing expectations about new candidates. As the Polish Wikipedia grows, the number of articles also grows. Articles are getting longer and more complex. Users gain more experience and have richer edits history. This paper contains some data and conclusions, which Spychała et al. [1] were unable to present due to limitations placed on length of the conference publication.

Administrators (or sysops) are very dedicated and trustworthy participants of the Wikipedia projects in all language versions. Thanks to community decision, they have received special privileges and use administrative tools to exercise preventive and policing functions. Administrators have the right to edit all the Wikipedia articles as well as many other privileges—understood rather as duties. These powers are not meant to give them editorial control over the project, but rather provide mentoring and technical assistance in other wikipedians' work. Administrators also serve by providing assistance, especially to newcomers, in editing of Wikipedia. All newly registered users get their guides—the administrators to whom they can always turn for help and be sure they will receive it as soon as possible.

Kittur et al. [4] claimed, that due to increasing amount of management work at Wikipedia, such as content quality control, coordination, maintenance, that are caused by the increasing popularity and amount of content in Wikipedia, the importance of administrators is increasing. This creates a potential risk that administrators may become overwhelmed by the amount of work and their response times become longer. Especially, Ortega et al. [5] showed that after peak of popularity of the Free Encyclopaedia, its user base growth slowed down. But in the same time, the amount of content is still increasing. If the Wikipedia is to keep its pace of growth, then some measures to increase its users base have to be taken.

However, the Polish community of Administrators is growing slower than expected; hence, the question whether and why this community shuts out candidates for new members. Currently, there are 149 administrators on Polish Wikipedia—for comparison, 1,147 administrators work currently on the English version. Of course, the English version is much more developed, but sheer number of people with administrative privileges is impressive. On the other hand, Ortega et al. [5] notes that the Polish Wikipedia has the highest rate of automated bots used to help administrators in their duties. This fact can explain why the number of administrators is not enough to prophesize doom of Polish Wikipedia.

Administrators are elected in a special procedure, the rules of which are clearly defined. This procedure is called Request for Adminship (RfA). As it was already mentioned, the privileges for administrators are granted by Wikipedia community, in a vote in which the right to vote is given to those Wikipedia users, who are well-known and respected members of the community and know and respect the established rules on the website. Wikipedians who are candidates for the administrator position must "have a minimum of 1,000 not deleted edits, first of which has to be made at least 3 months prior to the date of filing the candidacy". Nominations for administrator candidates are adopted by a special form on the web page that also contains the regulations and the list of candidates. New administrators are elected during a voting that lasts a week (168 hours). Wikipedians, in order to be allowed to vote, must have been registered for at least one month and have a minimum of 500 not deleted edits.

Interestingly, in case of English version of Wikipedia, no formal conditions are required in order to declare a candidacy for an administrator. The only conditions are possession of an account and trust among other users. Despite this, the page with the declaration forms contains the information that in case of self-nominating, it is recommended to have at least 2,000 edits for a minimum period of 3 months. Another important difference is that in the case of English version of Wikipedia, new administrators are elected not by voting, but by discussion. Moreover, "the consensus in RFA is not achieved by exceeding a threshold, but by the strength of the justification of the candidacy".

There is one aspect of this paper which distinguishes it from work presented by Spychała et al. [1], namely our analysis of lifespans of administrators. It allowed us to make sure that research, which is main topic of this paper, is well founded. What is more, analysis of lifespans of administrators allowed us to show analogy between our work and part of Ortega's work [5]. Moreover, one can clearly see correspondence between behavioural patterns exhibited by administrators and those exhibited by the core of most active users. Analysing sub-communities of user community of Wikipedia is challenging and interesting research direction, which we plan to take in our future research.

The rest of this work is divided as follows: in Section II, the related work is presented. Section III contains data motivating our research and base statistics, which show that growth of Polish-language Wikipedia Administrators group has slowed down. Data presented there is extended by data for year 2011, in comparison to data gathered by Jankowski-Lorek et al. [6]. In Section IV, Multidimensional Behavioural Social Network is used to analyse historical voting data. This analysis is the main contribution of this paper. Section IV also contains answer to question stated above. Section V presents conclusions and suggestions for future work.

II. Related Work

The problem of evaluation and recommendation of users requesting for adminship in Wikipedia has been addressed in several papers. In one of them, Burke et al. [7] try to indicate the features and qualities determinative for the user selection to the position of administrator. On the basis of publicly available tips for candidates [8], a set of attributes, that a future administrator should have has been developed. Behavioural data and comments, not page text, were used to evaluate candidates. Authors counted each candidate's edits in various namespaces (article, article talk, Wikipedia, Wikipedia talk, wiki projects, etc.) to calculate total contribution as well as contribution diversity. They also measured user interaction, mainly activity on talk pages, but also participation on arbitration or mediation committee pages and a few others. There are also several other statistics, but the ones mentioned seemed to be the most relevant to the candidate's success. Especially successful were candidates with strong edit diversity, mere edits in Wikipedia articles did not add much more chance of success. In user interactions, article talk page edits were the best predictor of success, with other authors talk page edits being rather poor. Burke et al. also confirmed Kittur's [9] results that the percentage of indirect work (coordination, discussion, etc.) grows over time, the share of articles in all Wikipedia edits is decreasing.

It is noteworthy that in Burke et al. [7] only the qualities of each user were evaluated. Leskovec et al. [10] have shown that the outcome of the voting depends on the candidate and his or her place in the community. They found out that the probability of one person's vote to be positive is correlated with the basic relative figures such as: who—voter or candidate has more edits, who has more barnstars (awards given by other Wikipedia users), the extent of collaboration of the two, etc. Authors strongly noted that the vote value (positive or negative) is not just a function of candidate, but both voter and candidate.

Kittur et al. [11], analysed the impact of the similarity of users on their mutual assessment. The examined data were collected from three websites: Wikipedia, Stack Overflow, and Epinions. The important feature of those websites is the possibility of mutual evaluation between their users. In case of Wikipedia it is the RfA voting. Two users were considered similar, when they have performed similar actions, which in case of Wikipedia were edits of articles. The authors concluded that, in case of Wikipedia, the possibility of casting a vote for a candidate increases with the increase of the similarity between the candidate and the voter. The voters, who are similar to the candidate, are less driven by the objective qualities (status in the community), such as experience in development of Wikipedia. Candidate's status determines casting a vote for that candidate when the voter and candidate are only slightly similar.

An interesting observation is that during the voting, there are much more voters similar to the candidate in a group with higher status than in a group with relatively lower status. This may suggest that during RfA the voters do not constitute a representative sample of community. This allows for the outcome of the election to be predicted when profiles and similarity of a few first voters and the candidate are known. To effectively predict the result of the voting one does not even need to know the votes given by the first voters.

The quality of Wikipedia articles depends on the level of cooperation of the editors. Rad et al. [12] decided to examine the history of article edits, and on that basis determine the mutual attitude of the editors and how controversial is the given article. Casting a vote during new administrator election was adopted as an indicator of relation between two users. If the voter has a positive attitude toward the candidate, the vote will be positive. In case of a negative attitude, the vote will be negative. The authors decided that the co-edit of the article is a pair of changes of the same section of the article, which were set apart in time by less than a fixed number of revisions. A social network with nodes labelled with users' profiles and directed edges labelled with users' co-edits, was also considered. This graph was used to induce a decision system and train a classifier, which was highly effective in predicting votes. What is important, is that this approach is complementary to the ones described earlier. It is based on the analysis of Wikipedia articles and their edit history and not on the aggregated statistics of the community. What is interesting, it turned out that it is relatively easy to predict positive votes. It seems that they are influenced by the most recent history of cooperation. On the other hand, the high quality of prediction of the negative votes required appropriately bigger and richer history of cooperation. The authors risked the statement that the users can remember disagreements for a long time and during a voting they can be guided by hidden qualities, like for example, the votes already cast in a given voting.

III. Statistics for Request for Adminship Procedure

During the period 2005 – 2011, the growth of Polish Wikipedia's content was steady. It can be seen on Figure 1, which presents an average number of new articles per day during examined period. Data for Figure 1 were obtained from freely available web-page presenting some statistics for the Polish Wikipedia [13]. On Figure 1, one can clearly see, that in the year 2006, there was a peak in pace of content growth, but in the following years the growth speed declined and steadied.

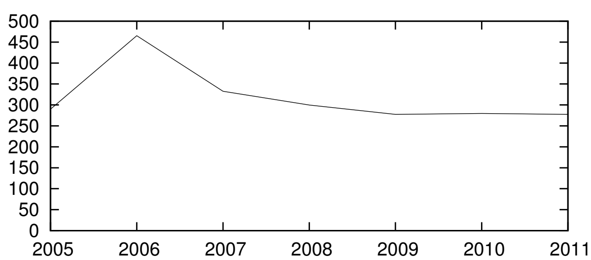


Fig. 1. The average number of new articles per day in each year

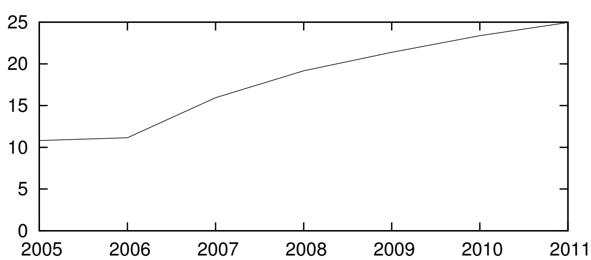


Fig. 2. The average number of edits per article in each year

In the same period, articles' content was getting more complicated. Because of that, there was increasing amount of management work, such as content quality control and maintenance. The increased articles' complexity can be approximated by an average number of edits per article. Values for averages for each year in examined period are presented on Figure 2. Data for Figure 2 were also obtained from above-mentioned web-page presenting some statistics for the Polish Wikipedia [13]. The above-mentioned figure shows, that the average number of edits per article was rapidly increasing during examined period. It indicates that amount of management work also increased, and leads to need for growth of the Administrators Community.

As of December 31, 2011 the Polish-language Wikipedia had 171 administrators. Since 2005, there were 307 votings on RfA. 177 of those ended with granting the administrator privileges to a candidate, in 110 of those, the candidates were rejected and in about 40 votings, the candidates resigned before the end of the voting and about 30 votings were cancelled (due to statutory requirements or lack of acceptance of the nomination by the candidate). About 38 administrators were chosen before the introduction of RfA procedures in March 2005. The data on RfAs do not sum up for several reasons. Among them are: verification votings and losing privileges by administrator either by giving them up or being revoked by the Arbitration Committee.

In the current version, the procedure states that a candidate for an administrator must have an account for at least three months and at least one thousand not deleted edits. In order to participate in the voting, user must have an account for at least 2 weeks and at least 500 article edits. Voting begins at the moment, when the candidates confirm, that they are willing to take the administrator position, as users can apply for the position themselves or be nominated by other users. In order to receive administrator privileges, the candidate need to receive at least 20 votes "for" and it must constitute of at least 80% of the sum of the votes "for" and "against". If the candidate does not receive the required number of positive votes or do not meet the formal requirements, he or she can apply again after at least 60 days since the end of last voting. A similar rule applies to the administrators who resigned from their position but would like to receive the privileges again.

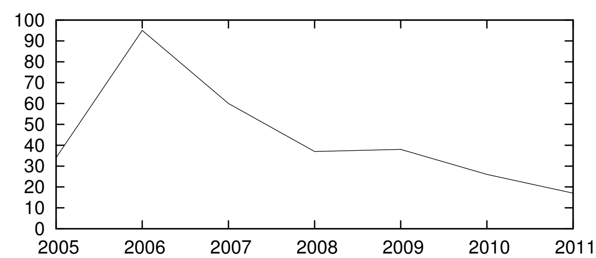


Fig. 3. Number of votings in each year

Figure 3 is presenting the number of votings in each year; the peak can be observed in the year 2006, when that number reaches 95, while a year before it reached only 34. One year after the peak, the number of votings dropped to 60. With the exception of the years 2006-2007, the number of votings never exceeded 38. In the years 2010-2011 that number declined below 34. The number of RfAs between year 2006 and 2011 decreased by nearly three quarters (form 95 to 26).

Fig. 4. The percentage of accepted RfA in each year

The percentage of the accepted nominations in each year (see Figure 4) can be divided into three periods. The first one consists of the years 2005-2008, when the percentage of the accepted candidates ranged from 57 to 70. The second period are the years 2009-2010, with the percentage below 50 (47% and 42% respectively). Between the years 2008 and 2010 the percentage of the positive RfAs fell by almost a half (from 70% to 42%). The third period, which accounts for the year 2011, is characterized by the relatively high number of positive RfAs. However, it should be noted that the number of the votings performed at that time was significantly lower than in the previous years.

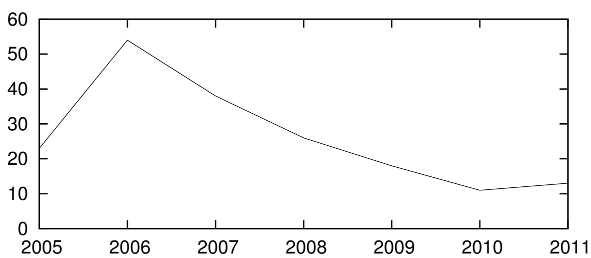


Fig. 5. The number of elected administrators in each year

The number of elected administrators is presented on Figure 5. The peak, as with number of voitings, can be observed in the year 2006, when that number reaches 54, while a year before it reached only 23. One year after the peak, the number of elected administrators dropped to 38. In the following years, number of elected administrators steadily went down to 11 in the year 2010 and 13 in the year 2011. With the exception of the year 2006 and the year 2007, the number of elected administrators never exceeded 30. In the most recent years, that number declined below 20.

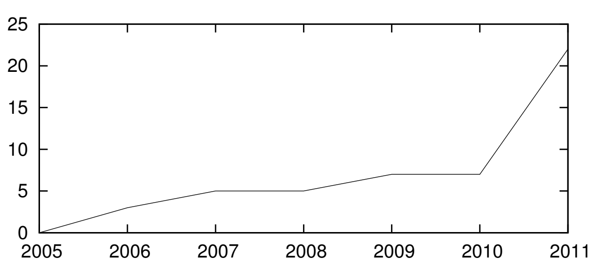


Fig. 6. The number of administrators who left the project in each year

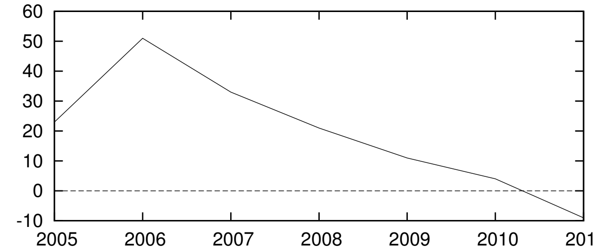


Fig. 7. The growth of the Administrator community in each year

The number of administrators who were relieved from duty is presented on Figure 6. There is steady upward trend in number of former administrators. The value in the year 2011 is equal to 22, which is probably a little too big. The errors are possible because our parsers for the Wikipedia's pages [14], [15] were not accurate enough. But the key fact is illustrated on Figure 7, which presents growth of number of Administrators of the Polish Wikipedia. The graph is an approximation of the real growth, because some face were not considered in data used to build it. One issue was with administrators who gave up their duties and were later re-elected. Since an administrator can leave the project and then come back again, in our analysis we used only the date of his first election to this function and the date of his last departure from the project. If some administrator has departed from the project and then was later re-elected, then his case lowered our approximation of growth of the Administrator Community of Polish Wikipedia by one. Fortunately, set of re-elected administrators has low cardinality. Because of that, our approximation although not accurate, can show overall trend. And it can be clearly seen, that after the peek of popularity of Polish Wikipedia in the year 2006, the above-mentioned trend in the growth of the Administrator Community of Polish Wikipedia is downward.

The next study, related to the experience of candidates prior to granting them administrator privileges, has been conducted on 97 users, who recently received them. In case of those elected before, the gathering of data was impossible because of gaps in the logs of Polish Wikipedia.

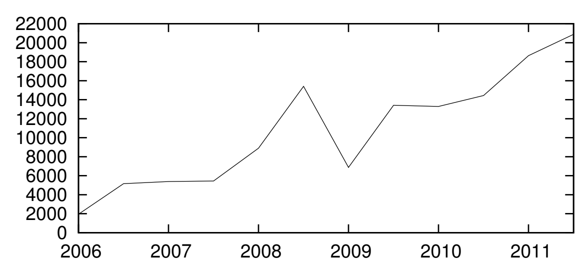


Fig. 8. The average number of edits made by user at the moment of receiving administrative privileges

One of the factors causing the most discussion during the voting is the number of edits performed by the candidate. RfA rules contain the following sentence: "Users who want to candidate for adminship (...) must have at least 1000 not deleted edits". Often, however, this number is considered by the voters to be too small. Basing on the analysis of the number of edits, it can be seen (Figure 8) that the minimum falls on the first half of 2006 with an average of 1,957 edits. This value then grows up to 2011 when it slightly exceeds 20,000 edits. This indicates that year by year, candidates needed to have greater experience in order to be accepted as administrators. The difference between the level of experience required by the regulations and the level widely accepted has been increasing as well. A similar phenomenon can be observed in the German Wikipedia, where—according to the voters—in the second half of 2010, candidates were accepted only if they had over 10,000 edits.

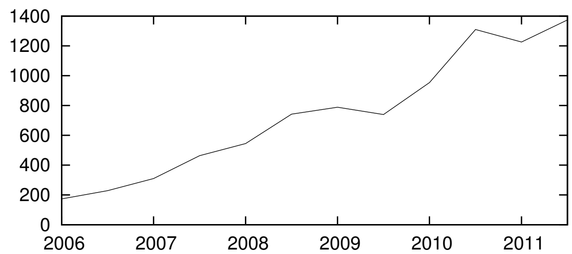


Fig. 9. The average number of days since creation of the account at the moment of receiving administrative privileges

Another factor that stirs up emotions at the time of voting is the seniority (understood as time since the first not deleted edit) on Wikipedia. The terms of voting set the following requirements: "Users who want to candidate for adminship (...) must have at least 1000 not deleted edits, the first of which took place at least 3 months before the date of candidacy proposition". The seniority (in days) of candidates, before the date of registration and acquiring the administrator rights, had been analysed (see Figure 9). This, however, is not exactly the same value as the required by the regulations. The measured seniority in the first half of the year 2006 was 173 days. This value has been gradually increasing: from 463 days in the second half of 2007, to 788 days in the first half of 2009, with a slight decline in the second half of 2009 (739 days). In the second half of 2010 the value reached 1310 days. This result, however, may be unreliable due to the fact that during that period only two votings took place. In the second half of 2011 the measured value reached 1374 days. The overall analysis of the chart shows that in the year 2006 candidates had less than a year of seniority, however, since mid-2008 the seniority is at least two years. The last two candidates who had less than one year of experience were selected in February 2009 and November 2008.

All three figures (Figure 3, Figure 4, and Figure 5) show the downward trend in the total number of newly appointed administrators between 2006 and 2011. This decrease gives reason for serious concern as the amount of required administrative work on Wikipedia is constantly growing. This phenomenon may have several possible explanations. The first explanation is the declining number of candidates who accept their nominations for administrators (that would explain the decreasing number of RfA votings), but the confirmation of this hypothesis is beyond the scope of this paper. Nevertheless, related works have shown that in recent years Wikipedia has experienced a downtrend in the amount of user contributions, which reflects the general decline in motivation as shown by Suh et al. [16].

The second explanation states that the number of positive nominations decreases due to the changing criteria for selecting and accepting candidates. Those criteria can vary in many ways; however, our research shows that they are connected to the candidate's experience. This experience can be initially estimated on the basis of the edits performed, but the more accurate measurement (presented by Burke et al. [7]) represents the number of article edits in a specific category.

The more damaging prospect is the fact that the administrator community is chosen on the basis acquaintance between current administrators and candidates. The next section discusses, if that is the case.

IV. Acquaintance in the administrators society

Acquaintance is a social phenomenon, which can lead to shutting out new members of some society. To approximate it, we used implicit social connections contained in history of talk pages in Wikipedia. We claim, that connections obtained in above-mentioned way, are indeed good approximation of acquaintance relation. Jankowski-Lorek et al. [6] proven mentioned claim to be true.

A. Data description

Data and multidimensional behavioural social network used for this paper were gathered, aggregated and made available by the team led by dr. Adam Wierzbicki. Methodology, data and networks are described in greater detail by Jankowski-Lorek et al. [6]. Examined period encompasses the years 2005-2011.

Basically, the network consists of four dimensions:

- Co-edits,
- Reverts,
- Discussion,
- Topics.

Weights in co-edits dimension are based on number of words written by one author next to the text written by some other one in the text of articles. The authorship information for a particular fragment of text was obtained by analysing its first occurrence in the whole edit history of examined page.

Edge strength in reverts dimension is based on the number of edits made by one author and reverted by other. It was obtained by searching identical revisions before the examined one. If it was found, each pair of examined revision and revisions after the other identical one was used to calculate number of reverts.

Similar to co-edits, edge strengths in discussion network were stated as number of words written by one author next to text created by other one. But in this dimension, the talk pages were considered.

The last dimension, topics, was a little different to other ones. It was a bipartite graph connecting authors with categories in which they have edited at least one article. The edge weight was exactly the number of article edits made by given author in the particular category.

One of the most important observations made by Jankowski-Lorek et al. [6], is that discussion network can be interpreted as social relation of acquaintance. Jankowski-Lorek et al. [6] conducted another research, a survey among Polish Wikipedia users. However, interpretations of other dimensions have not been confirmed.

The data contained two more graphs: positive votes network and negative ones. If, during RfA procedure, user has cast positive vote for candidate, then an edge in the positive votes network has been created. Its weight was equal to number of positive votes cast by the user for the candidate. Weights of more than one were possible only if the user was a candidate more than once. Network of negative votes has been created in an analogous manner, but taking the negative votes instead.

Turek et al. [3] and Jankowski-Lorek et al. [6], intersected each dimension with positive and negative votes networks, in order to examine correlation between social network dimensions and RfA votings. Both graphs were analysed separately and features distinguishing them have been found.

Research presented in this paper studies only on the discussion dimension. The reason for such decision is that discussion network can be interpreted as a real relation—acquaintance. For each year, graph of discussion network has been intersected with positive and negative votes networks. Maniu et al. [17] suggest using one, signed network, especially, when there is a strong correlation between both networks as shown by Leskovec et al. [18]. Two separate graphs were used for two reasons:

- To maintain consistency with analysis presented by Turek et al. [3] and Jankowski-Lorek et al. [6],
- To separately check positive and negative impact on RfA procedure of acquaintance relation.

Ortega et al. [5], defined the core of the Wikipedia user community as a group of most active editors in analysed month. Most of the Wikipedia content is created by the group of core users. The core existed in all examined language versions of the Wikipedia, in particular in Polish and English versions, but the behavioural patterns of users were different in both language versions. The first indication of that fact are the differences in the structures of core user groups of both Wikipedias. To name just one, in the English version new Administrator is elected by reaching consensus among the users fulfilling a set of requirements, while in the Polish version – by "formal" voting within similar group. Ortega notes, that structure of the core user group of Polish Wikipedia is different than structures of similar groups of other top ten Wikipedias.

The other difference between social behaviour in case of users of the Polish Wikipedia and the English one is what we could call the survival patterns of the core users. Although in all top ten Wikipedias users entered to the core relatively quick – after less than a half year, the stability of a core group's composition was quite different. In case of the English Wikipedia, users forming the core often left it after a month. In case of the Polish version of the free encyclopaedia, users stayed in the core for three months. But in both cases there exists part of the core formed by users who stay there for years.

There is also a difference between commitment level of authors after leaving the core group. Users of the English version of the Wikipedia often leave the project, after leaving the core group. In case of the Polish version, this is not the case. Polish users show strong commitment to the project. After leaving the core, for long period they often revise articles and sometimes come back to the core.

TABLE I. STATISTICS FOR LIFESPANS IN MONTHS FOR ENGLISH WIKIPEDIA

	active	former	combined
min	28	2	2
max	120	40	120
average	75.68	12.73	72.36
median	73	13	73

One can argue, that the rotation of Administrators is high enough to prevent from forming closed groups based on acquaintance. But to decide if such claim is well founded, it is necessary to describe the commitment part of behavioural patterns in case of the Administrators of the Wikipedia. The intuition is that, the Administrators are a group of some of the most active users of the Wikipedia and will show behavioural patterns similar to members of the core users group. But the question is whether it will be more similar to the English core group or to the Polish.

In order to answer above-mentioned question, we analysed list of Administrators of the English Wikipedia [19] and list of Administrators of the Polish Wikipedia [14], [15] and gathered statistics for lifespans of Administrators for mentioned Wikipedias. In our research, *lifespan* of administrator is the length of period, in which user has been granted administrator privileges. If user resigned and was later appointed for being administrator one more time, then his lifespan is period from first election to the last resignation.

For each Wikipedia, we selected three sets of Administrators: all active Administrators, all former Administrators and a combined set of all Administrators (active and former ones). Within each set we calculated lifespans of its members. For obtained lifespans we calculated: minimum, maximum, average and median. Lifespans and statistics relate to events up to the end of the year 2011. Lifespans of Administrators, who were active after the year 2011, were calculated to the 1^{st} of January 2012. As the unit of lifespan we have chosen one month. In case if there were some remaining days, lifespan was rounded to the nearest month. It is worth noting, that in all cases average lifespan and median are almost equal to each other. Because of that, we will analyse average lifespans keeping in mind, that our conclusions are also backed up by medians.

Statistics of lifespans of Administrators of the English Wikipedia are presented in Table I. It is worth noting, that there is large difference between lifespans of active and former Administrators. Active Administrators are active for at least 2 years. They remain active for about six years on average. The maximum length of practice in the case of an active Administrator is ten years. On the other hand, most former Administrators are active for about a year on average. Former Administrator with the longest service was on duty for more than three years. Because of such short average lifespan of former Administrators, one could not infer about possibility of forming strong groups based on acquaintance. The other fact, which is noteworthy, is that the group of Administrators

TABLE II. STATISTICS FOR TIME OF SENIORITY IN MONTHS AFTER BEING FREED FROM ADMINISTRATORS DUTY FOR ENGLISH WIKIPEDIA

	time of seniority
min	0
max	120
average	50.36
median	48

TABLE III. STATISTICS FOR LIFESPANS IN MONTHS FOR POLISH WIKIPEDIA

	active	former	combined
min	7	1	1
max	79	75	79
average	46.57	38.96	44.08
median	45	40	44

of the English Wikipedia yields similar behavioural patterns to the its core.

In Table II statistics for time of seniority in months after being freed from administrators duty for English Wikipedia are presented. Data for Table II was obtained from [20]. There are some cases when administrator lost his privileges and left the project. But on average, after quitting administrators group, users tend to support the free encyclopaedia. And it is substantial support as a median of time spent in project after quitting administrator role is about four years. It is noteworthy difference from the core users described by Ortega et al. [5] and should receive more attention in future studies.

Table III presents statistics for lifespans Administrators of Polish Wikipedia. The first thing to note is, that in case of the Polish Wikipedia maximum and average lifespans are quite similar for active and former Administrators. The longest service of active Administrator is more than six years. And it is also length of the longest service of former one. Both, active and former Administrators will be on duty for three to four years on average. Is is a period which length is comparable to length of the whole examined period (years 2005–2006). This, in contrast to the English Wikipedia, leads to concern if there is possibility that some kind of ”elite” in the Administrator community of the Polish Wikipedia is forming. Another fact is that, the Administrator community of the Polish Wikipedia shows similar behavioural patterns to the core of Polish Wikipedia and is quite different from the Administrators community of the English Wikipedia. Unfortunately, due to the time constraints, we were unable to ensemble table with statistics for time of seniority in months after being freed from administrators duty for Polish Wikipedia. It is our priority to gather data and compare it to corresponding data for the English version.

B. Base statistics

In order to compare graphs resulting from intersecting the discussion dimension graph and votes nets for each year, base statistics were obtained for edges' weights. The used measures were: minimum, maximum, arithmetic mean, median, first and third quartile.

For each graph, i.e., discussion network, discussion intersected with positive and discussion intersected with negative votes in each year, empirical distribution functions were calculated. Distribution graphs for examined years are presented in Figure 10 and Figure 11. Values of x-axis are logarithms

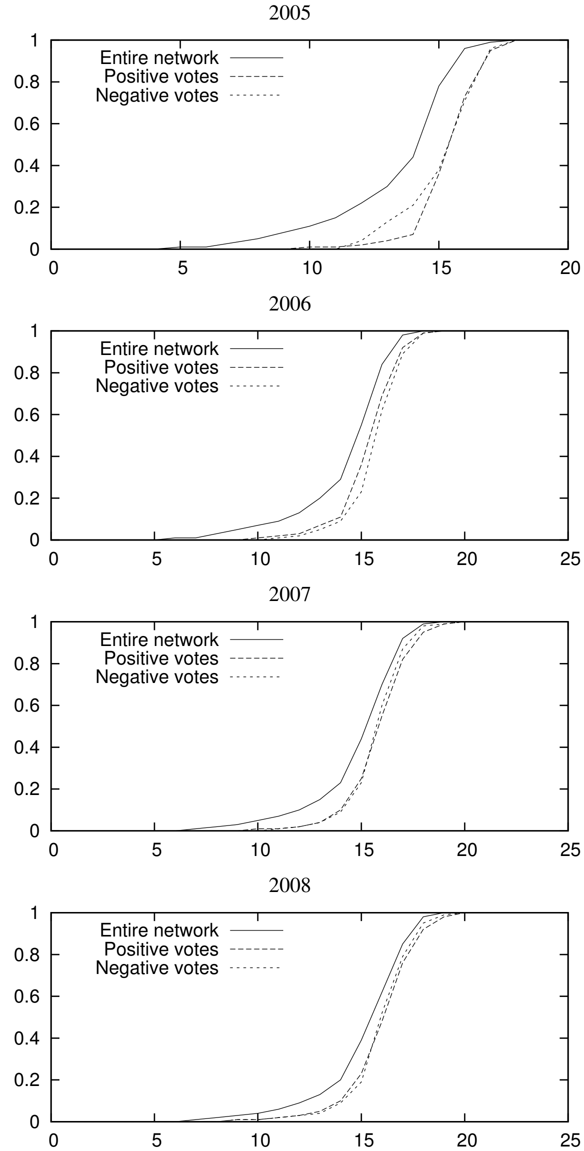


Fig. 10. Distributions of discussion network edge strengths

of edge strengths. Since 2007, the distribution of data is analogous to that described in the article. Both arithmetic mean and median are significantly higher for positive votes. As a result, it can be concluded that the Wikipedia user community is developing steadily.

The stability of the development of the Wikipedia administrator community is also reflected by the empirical cumulative distribution charts. The shapes of the curves are similar, so it can be concluded that the probability distributions describing different parts of the Wikipedia user community originate in the same distribution family. This means that the behaviour of the voters is not subject to sudden changes, but at most it undergoes a calm evolution.

C. Clustering coefficients

Clustering coefficient is a measure of degree to which nodes in a graph tend to be clustered together. The global version, which is used in this article, was designed to give an

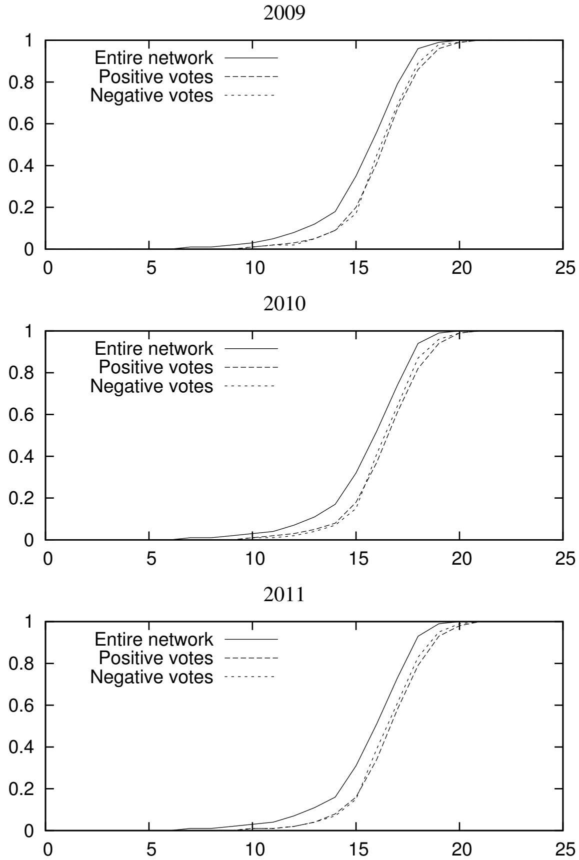


Fig. 11. Distributions of discussion network edge strengths

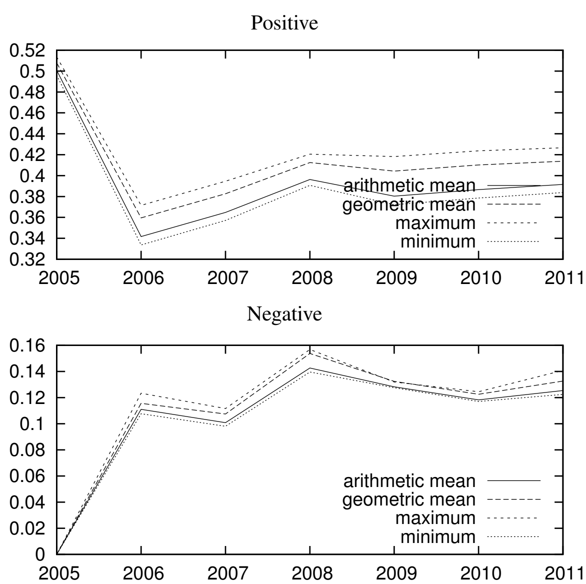


Fig. 12. Clustering coefficients

overall indication of the clustering in the network. Basically, for undirected graphs, it is a ratio between number of closed triplets (three nodes connected by two links) to number of all triplets (three nodes connected by either two or three links).

For directed, weighted graphs a generalization was proposed, it is described in detail Opsahl et al. [21]. Opsahl et al. proposed four measures to calculate triplet value:

- Minimum of edges' weights (mi),
- Maximum of edges' weights (ma),
- Arithmetic mean of edges' weights (am), and
- Geometric mean of edges' weights (gm).

The intuition is as follows: the minimum version is used to find the weakest group in graph, the maximum to find the strongest. Both means give an indication of the strength of ordinary clusters. Opsahl et al. [21], [22] also created tnet library [23] for R software [24].

For each year from 2005 to 2011, the clustering coefficients were obtained for intersections of acquaintance networks with graphs of positive and negative votes. Those coefficients are presented in Figure 12. There are four values (calculated for each of the measures mentioned before) for both graphs.

A few facts can be observed. The first is that there are no very weak or strong groups in Polish Wikipedia society. There is no “elite”, which governs RfA procedure or has taken over the administrator society and has power to rule Polish Wikipedia.

The second fact is that clustering coefficients are relatively low and their growth rate is low and negligible. We argue, that decrease in successful administrator elections is not a result of a building up acquaintance relation. Voters do not cast positive votes for their acquaintances or cast negative votes for strangers. The anomaly in year 2005, that clustering coefficients have abnormal values, is most likely caused by the fact, that data for year 2005 were not complete.

V. Conclusions and Future Work

This paper presented the analysis of the development dynamics of the community of administrators of Polish Wikipedia. We have used multidimensional behavioural social networks as a tool to model relationships between wikipedians. The aforementioned analysis included examination of the community in each year from 2005 to 2011 as well as the analysis of the social network corresponding to the final state of the community. The analysis was based on the data from public Wikipedia data dumps.

The fundamental question which we sought the answer to was: “Is the administrator community of the Polish Wikipedia closing up?” It turns out that the answer is not straightforward and it depends on what aspects of the problem one put the greater emphasis, or how to define the “closing up” society.

The conducted analysis of the social network allows us to draw conclusions about the impact of the social system on the nominations of the new administrators. The results of this analysis clearly show that this phenomenon does not exist in the Polish Wikipedia. This is one of the arguments for the statement that the community of the administrators is not shutting out candidates. The administrator community is open to new members in the same way as it was in the beginning of the Polish-language Wikipedia.

However, the pace of growth of the administrator community is lower than it could be expected in case of a young and dynamically growing society. In the early years of development the number of votings was much higher than in the recent years and the number of new administrator appointments strongly declined. That could indicate, however, that the community is closing up after all.

Slower pace of growth and acceptance of new members can be caused by various factors. One such factor may be higher entrance requirements for candidates. Both administrators and regular editors of Wikipedia continue to develop and gain experience in new areas. At the same time, the history of their activity is freely available. For that reason, new users may have trouble with showing equally high achievements and contribution to Wikipedia development. This can be interpreted as closing up of the community by making prohibitive requirements for the new candidates, or as a kind of professionalization aiming to increase the substantive level of the Polish Wikipedia.

It is also possible, that potential candidates do not request for administrator's privileges or withdraw their own candidature from voting because they assess their achievements and contribution to Wikipedia development as not high enough to succeed as an administrator. That would lead to decreasing number of new candidates and appointed new administrators. In our opinion, such case is extremely hard to distinguish from other ones. The best thing to do for the Wikimedia foundation is to do social campaign among users of Wikipedia promoting administrator's duties and boosting confidence in users. This could lead to weakening self censure among candidates.

Our conclusion is that it cannot be claimed with certainty that the Polish Wikipedia community is shuttig off new candidates for administrators. We believe that the increase in the requirements of the current administrator community and users entitled to speak during RfA process toward administrator candidates stems from the community's desire to raise the quality and ensure maximum involvement of all the administrators in the development of Wikipedia.

Our research leads to another interesting conclusion. In Section IV, we have shown the statistics concerning lifespans of the Administrators of the Polish Wikipedia and the English one. Conclusions about the Administrator communities, which can be drawn from above-mentioned statistics, are coherent with ones presented by Ortega et al. [5] about core users of Wikipedias. Namely, users of the Polish Wikipedia will support the free encyclopaedia longer than their English counterparts. It is promising research direction to check if sub-communities of Wikipedias' communities are in some sense fractal. To check, if all sub-groups of users share similar behavioural patterns determined by language version of the project.

The results presented in this study describe the community of Polish Wikipedia administrators only partially. Further research should focus on the detection of new relations between the users and social networks associated with them. It is important to find methods that will allow the development of community to be automatically analysed on the basis of widely available data. The multidimensional behavioural social networks seem to be an ideal tool for this purpose. Richer description of the community could help predict the direction of its development, which may result in the early identification of threats. This will give the opportunity to counteract those threats and ensure the correct development of Wikipedia.

User community of Polish Wikipedia—in contrast to other language versions—is relatively little known and researched, although, it is an ideal subject for researchers dealing with social informatics. It can be an interesting subject for two types of research: new research, previously not conducted on such a social group, and repeated research, taken from a different version of Wikipedia and performed on the Polish version in order to compare the results and draw conclusions on the development of the latter in comparison to other versions.

Tools used to create multidimensional behavioural social networks for Polish Wikipedia were unable to create such graph for larger instances, e.g., English one. In order to conduct comparative research, scalability problems should be addressed. There is also possibility, that more scalable algorithms can be made on-line. This can allow development of on-line recommendation algorithm for RfA votings.

Acknowledgement

Authors of this paper thank the team led by dr. Adam Wierzbicki from Polish-Japanese Institute of Information Technology in Warsaw for sharing processed Polish Wikipedia data and multidimensional behavioural social network of Wikipedia users.

The second author was supported by the Polish National Science Centre grant DEC-2011/01/B/ST6/03867.

References

[1] J. Spychała, P. Turek, and M. Adamczyk, "Does the acquaintance relation close up the administrator community of polish wikipedia? analysing polish wikipedia administrator community with use of multidimensional behavioural social network," in *COLLA 2013, The Third International Conference on Advanced Collaborative Networks, Systems and Applications*, 2013, pp. 28–33.

[2] (2014, 05) The third international conference on advanced collaborative networks, systems and applications (colla 2013). [Online]. Available: http://www.iaria.org/conferences2013/COLLA13.html [Accessed: 20.05.2014]

[3] P. Turek, J. Spychała, A. Wierzbicki, and P. Gackowski, "Social mechanism of granting trust basing on polish wikipedia requests for adminship," in *SocInfo*, ser. Lecture Notes in Computer Science, A. Datta, S. W. Shulman, B. Zheng, S.-D. Lin, A. Sun, and E.-P. Lim, Eds., vol. 6984. Springer, 2011, pp. 212–225.

[4] A. Kittur, B. A. Pendleton, B. Suh, and T. Mytkowicz, "Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie," *World Wide Web*, vol. 1, no. 2, p. 19, 2007.

[5] J. F. Ortega Soto, "Wikipedia: A quantitative analysis," 2012.

[6] M. Jankowski-Lorek, L. Ostrowski, P. Turek, and A. Wierzbicki, "Modeling wikipedia admin elections using multidimensional behavioral social networks," *Social Network Analysis and Mining*, pp. 1–15, 2013. [Online]. Available: http://dx.doi.org/10.1007/s13278-012-0092-6 [Accessed: 15.05.2013]

[7] M. Burke and R. Kraut, "Taking up the mop: identifying future wikipedia administrators," in *CHI Extended Abstracts*, M. Czerwinski, A. M. Lund, and D. S. Tan, Eds. ACM, 2008, pp. 3441–3446.

[8] (2014, 05) Wikipedia: Guide to requests for adminship. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Guide_to_requests_for_adminship [Accessed: 20.05.2014]

[9] A. Kittur, B. Suh, B. A. Pendleton, and E. H. Chi, "He says, she says: conflict and coordination in wikipedia," in *CHI*, M. B. Rosson and D. J. Gilmore, Eds. ACM, 2007, pp. 453–462.

[10] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, "Governance in social media: A case study of the wikipedia promotion process," in *ICWSM*, W. W. Cohen and S. Gosling, Eds. The AAAI Press, 2010.

[11] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, and J. Leskovec, "Effects of user similarity in social media," in *WSDM*, E. Adar, J. Teevan, E. Agichtein, and Y. Maarek, Eds. ACM, 2012, pp. 703–712.

[12] H. S. Rad, A. Makazhanov, D. Rafiei, and D. Barbosa, "Leveraging editor collaboration patterns in wikipedia," in *HT*, E. V. Munson and M. Strohmaier, Eds. ACM, 2012, pp. 13–22.

[13] (2014, 05) Wikipedia statistics polish. [Online]. Available: http://stats.wikimedia.org/EN/TablesWikipediaPL.htm [Accessed: 20.05.2014]

[14] (2014, 05) Wikipedia:administratorzy. [Online]. Available: http://pl.wikipedia.org/wiki/Wikipedia:Administratorzy [Accessed: 20.05.2014]

[15] (2014, 05) Wikipedia:przyznawanie uprawnień/przyjete. [Online]. Available: http://pl.wikipedia.org/wiki/Wikipedia:Przyznawanie_uprawnie%C5%84/Przyj%C4%99te [Accessed: 20.05.2014]

[16] B. Suh, G. Convertino, E. H. Chi, and P. Pirolli, "The singularity is not near: slowing growth of wikipedia," in *Int. Sym. Wikis*, D. Riehle and A. Bruckman, Eds. ACM, 2009.

[17] S. Maniu, B. Cautis, and T. Abdessalem, "Building a signed network from interactions in wikipedia," in *DBSocial*, D. Barbosa, G. Miklau, and C. Yu, Eds. ACM, 2011, pp. 19–24.

[18] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, "Signed networks in social media," in *CHI*, E. D. Mynatt, D. Schoner, G. Fitzpatrick, S. E. Hudson, W. K. Edwards, and T. Rodden, Eds. ACM, 2010, pp. 1361–1370.

[19] (2014, 05) User:noseptember/list of administrators. [Online]. Available: http://en.wikipedia.org/wiki/User:NoSeptember/List_of_Administrators [Accessed: 20.05.2014]

[20] (2014, 05) Wikipedia:missing wikipedians. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Missing_Wikipedians [Accessed: 20.05.2014]

[21] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social Networks*, vol. 31, no. 2, pp. 155–163, May 2009. [Online]. Available: http://dx.doi.org/10.1016/j.socnet.2009.02.002 [Accessed: 15.05.2013]

[22] T. Opsahl, *Structure and Evolution of Weighted Networks*. University of London (Queen Mary College), London, UK, 2009. [Online]. Available: http://toreopsahl.com/publications/thesis/ [Accessed: 15.05.2013]

[23] (2013, 05) tnet: Analysis of weighted, two mode, and longitudinal networks. [Online]. Available: http://toreopsahl.com/tnet/ [Accessed: 15.05.2013]

[24] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org [Accessed: 15.05.2013]

An Analysis of the Generative User Interface Pattern Structure

Stefan Wendler and Detlef Streitferdt
Software Systems / Process Informatics Department
Ilmenau University of Technology
Ilmenau, Germany
{stefan.wendler, detlef.streitferdt}@tu-ilmenau.de

Abstract — **Current business information systems extensively rely on graphical user interfaces (GUIs). These sub-systems enable the interaction between the end user and application kernel services that are essential for the business process instances. Due to dynamic and rapid changes of both business processes and their required services, a strong need for the quick adaptation of GUIs to the occurring changes arose. As both efficiency and usability are essential for the GUI adaptation, model-based development processes that involve patterns and their instantiation for specific GUI contexts have been suggested by ongoing research. Being based on human computer interaction patterns, the new kind of pattern needs to be formalized in order to enable the automated processing of configurable instances by generator tools. However, current research is still at the edge to express the concepts for such generative user interface patterns. Crucial factors and impacts of those patterns have not been described sufficiently yet so that a standardized format for the expression of variability is still missing. With our work, we briefly review the current state on modeling user interface patterns and their requirement aspects. The ultimate objective of this paper is the development of an analysis model that is able to express both the structure and variability concerns of user interface patterns in detail. To evaluate and illustrate the analysis model concepts, selected user interface pattern instances are modeled via object models. As result, a detailed description of generative user interface patterns is achieved, which can be applied as a basis for the verification of recent approaches of model- and pattern-based GUI development or even the synthesis of a dedicated user interface pattern language.**

Keywords — user interface patterns; model-based user interface development; HCI patterns; user interface generation; graphical user interface.

I. INTRODUCTION

A. Motivation

Domain. Business information systems of our days are being maintained to upkeep or raise their effectiveness in supporting users carrying out operative tasks, which are demanded by the business processes of the respective company. Being a layer of a given business information system, the graphical user interface (GUI) is part of a value creation chain, as it enables the user to access functional, data and application flow related components of sub-systems located lower in hierarchy. Accordingly, the GUI allows the user to select and initiate functional behavior that processes data relevant to active tasks. As result, value is being created, which is meaningful to the sequence of the business process within the value creation chain. Since systems are constantly matched closer to the set of tasks of the business processes, users are facing an increase in task scope and complexity. Ultimately, the need for well designed, adaptive and easy to maintain GUIs has emerged.

This is a revised and substantially augmented version of "An Analysis Model for Generative User Interface Patterns", which appeared in the Proceedings of The Fifth International Conferences on Pervasive Patterns and Applications (PATTERNS 2013) [1].

GUI requirements. In this context, a user interface primarily is required to fulfill both the criteria of functionality and usability. On the one hand, a GUI has to reflect the current process definition, and thus, offer access to the respective activities in order to provide effective support for the user. On the other hand, for this support to be efficient, the non-functional requirement of usability, which embraces the suitability for the task and learning, as well as a high degree of self descriptiveness [2], plays an important role for testing and the acceptance for productive runs.

GUI adaptability. As business processes tend to change over time, the functional requirements based on them, such as use cases or task models, may change considerably, too. With those changes taking place, new requirements, having a significant impact on the GUI artifacts, are being introduced. Consequently, this part of the system has to conform to a high demand on adaptability besides the first release-specific requirements. Especially standard software systems, which offer a configurable core of functions to support business models, like applied in e-commerce, see a distinctive demand for adaptive user interfaces [2][3]. Accordingly, a user interface of a business information system has to be based on a software architecture or development process, which facilitates the transition to new visual designs, dialogs, interaction designs and flows without causing significant costs in manpower and time.

Current limitations. Nowadays, the above mentioned requirements still cannot be accomplished fully by automation and generative development processes. On the one hand, available GUI-Generators can only cover certain stereotype parts of the user interface and may not lead to the desired quality in usability [3][4]. On the other hand, model-based development processes, which are able to generate more sophisticated user interfaces, also cannot support all variations on interaction and visual designs the changing business processes may demand for [5]. Finally, concepts that combine increased reuse and automation in user interface development and adaptation are being sought of.

User Interface Patterns. Together with other researchers [2][4][6][7][8][9][10][11], we believe that certain aspects of the GUI can be modeled independently in order to be composed and instantiated to their varying application contexts. As evolution and individualism in GUI implementations generally induce high efforts, an approach has to be followed, which enables a higher degree of reuse, and hence, allows for more common basic parts to be shared among components. For reuse, the basic layout of a dialog, its positioning of child elements and navigation flow as well as reoccurring user interface controls (UI-Controls) and their data type processing are to be mentioned as candidates for automated generation. In this context, the occurring variability needs to be expressed by new artifacts in the development process chain. The need for a systematic description of reusable GUI artifacts arose and initially has found its expression in human computer interaction (HCI) [12][13][14] or, more recently, in user interface patterns (UIPs) [7][15]. In this regard, UIPs describe the common aspects of a GUI system in an abstract way. The developers concretize them with the required parameter information suited for the context of their instantiation.

UIP conception. The existing work about UIPs applied in model-based development processes [8][9][10] has laid down conceptual basics and milestones towards experimental proofing. However, no dedicated pattern definition for user interface development [6][16] has emerged yet, and so, the motivation of the PEICS 2010 workshop is still of high relevance [17].

Factor model. To progress towards a more detailed and complete UIP conception, we intensely elaborated requirements with impacts to architecture, formalization and configuration of UIPs in reference [5]. A process, which enables the instantiation of UIPs and their compositions to form a GUI of high usability and adaptability, altogether, needs such a clear basis of requirements. However, the factors we have modeled reside on a descriptive level that is not favorable to be directly translated to notations or formats for generative UIPs.

B. Objectives

The results of our work on the factor model in reference [5] have led us to the strategy to specify an analysis model for the UIP aspects and their various impacts. This model shall serve as a medium to close the gap between descriptive requirements of the factor model and formal notations. With the analysis model, we are detailing the requirements even more and progress towards a semi-formal notation for their description. The analysis model is intended to capture all essential aspects, properties and required parameters for context-specific application of UIPs. With this contribution, an initial version of the analysis model is presented.

We focus on the UIP representation and not its mapping or deployment process, since other researchers have advanced in that area, but still lack a proper UIP representation. This representation is elaborated here along with related work, criteria, examples and finally an analysis model. The following questions shall be answered by our analysis model:

- What information is needed to describe a UIP as a generative pattern applicable as a GUI architecture design unit?
- What elements a formal language has to feature in order to permit the full specification of such UIPs?

C. Structure of the Paper

The following section provides an overview of the pattern type to be covered in this work. To begin with, origin and basic definition of UIPs are presented with the aid of related references from the human computer interaction community. To address possible formalizations of UIPs, XML based languages, which enable the platform-independent specification of GUIs, are introduced. In addition, an UML-based approach that promises formal modeling of UIPs on the basis of class models is briefly described as well.

In Section III, we present an overview of the role UIPs may assume with respect to the development of GUIs in the domain of business information systems. In addition, a UIP based development and modeling concept is briefly introduced to inspire a comprehensive view on UIPs. Lastly, requirements related to UIPs are reviewed to draw a distinction to common user interface development practices.

In Section IV, the problem statement is formulated. We summarize the outcomes of our previous work on the examination of model-based development processes and valuate the current state of related work.

The description of our approach follows in Section V. Our main achievement is the elaboration of the analysis model that is presented in Section VI. Object models that are presented in Section VII will reveal additional details of the analysis model applied to UIP examples. Therefore, the object models will evaluate the applicability of the analysis model. The results of our work are reflected in Section VIII, before we conclude and suggest future work in Section IX.

II. RELATED WORK

A. Human Computer Interaction Patterns and User Interface Pattern Definition

To open the discussion of reusable GUI entities, aspects of patterns related to GUI development are now introduced. We approach the term “user interface pattern” (UIP), which will drive the further elaboration of related work. For this purpose, we ask what the origins for definitions of UIPs in the context of GUI generation are.

HCI pattern ambitions. The early stages of patterns for user interfaces were determined by the goal to describe reoccurring problems and feasible solutions for GUI design offering high usability. Borchers [14] stated that human computer interaction (HCI) experts had a hard time communicating their feats in ensuring a good design of a system’s GUI to software engineers. Thus, the idea was born to express good usability via patterns as this was already a good practice for software architecture design. In this regard, Van Welie et al. [18] argued that patterns are more useful than guidelines for GUI design. In addition, they suggested the term pattern for user interface design along with criteria how to assess the impact on usability of each pattern.

Research into HCI patterns went on and culminated into pattern languages such as the one created by Tidwell [19]. Prior to this development, Mahemof and Johnston [12] outlined a hierarchy of patterns, what already implicated that there are complex relationships inside HCI pattern languages.

No unified pattern notation. Some years later, Hennipman et al. [20] claimed that available HCI pattern approaches could be improved as there were still obstacles for their efficient usage. Their analysis of relevant sources revealed major issues such as the missing guidelines how to formulate new HCI patterns, integrate them in tools and how to apply them. The request for a standard pattern specification template already was formulated in references [14] and [18]. In this regard, Borchers mentions early sources adopting the famous pattern notion by Christopher Alexander. Finally, Fincher introduced PLML [21] in reference [22]. However, the issue of a missing standardized pattern format still persists [17], which eventually is detailed by Engel et al. [23]. Therein, they analyze the shortcomings of current HCI pattern catalogs, the intended standard notation of PLML and its extensions.

UIP definition. Vanderdonckt and Simarro [24] separate two main representations of patterns based on the intended usage. Descriptive patterns serve a problem description and solution specification purpose. In contrast, generative patterns feature a machine readable format as they are to be processed by tools and in particular GUI generators. Besides this rather general segregation, we have not found any elaborate definition on UIPs.

B. Formal Languages for GUI Specification

Now, we ask if there are languages available that may permit the formal specification of UIPs.

In our previous work [3][15], we already went into the possibilities to express UIPs with the means of mature GUI specification languages UIML [25] and UsiXML [26]. As these languages are focused on platform-independent full-fledged GUI specification and intended to be machine processed, some of their elements may be candidates to be included in a sophisticated UIP definition model. Both languages feature common elements to define the visual layout, interactive behavior and content of a certain GUI part. For pattern-specific application, UIML and UsiXML differ in their capabilities: UIML incorporates elements for template definition and a peer section, which decouples structures or UI-Controls within the layout from their technical counterparts. In contrast, UsiXML is based on a more complex approach, which defines a metamodel consisting of a model hierarchy and methodology [27]. The abstract (AUIM) and concrete user interface model (CUIM) may be of relevance for our objective.

C. UML Class Based Modeling of User Interface Patterns

In our search for UIP aspects and definitions we discovered an approach towards UIP modeling that relies on UML. No exact UIP definition was provided either but on the basis of given examples the individual UIP aspects were outlined rather clearly.

The UML is a common basis for modeling software systems. As a notation it is present in major CASE tools and is applied to express multiple aspects and views of a system in one comprehensive model. To further complement the aspects of a system in this model, an approach for modeling UIPs with UML class models was developed by Beale and Bordbar [6].

Common motivation. Their motivation is sourced from several problems. Firstly, they support our claim from Section II.A that no standard specification for UIPs does exist. Secondly, available UIP catalogs or collections [19] [28][29][30] vary in structure as well as their pattern relations, so that developers would need considerable expertise to use those resources effectively or train new development team members. The problem stated by Beale and Bordbar is that no uniform principles for searching and identification of suitable patterns for a given context can be relied upon to raise effectiveness. This applies to the comparison of patterns between existing catalogs as well. Thus, pattern languages did not provide support for comparison between alternative patterns suitable for the same context and their trade-offs. In the end, the developer would be faced with a multitude of available options to select UIPs for the context or system in focus.

UML approach. As a solution for both problems, Beale and Bordbar follow the idea to express UIPs by the same means as used for the system model. In their approach, a tool reads a UML system model and suggests appropriate UIPs for GUI implementation or refinement. As input, the pattern matching tool analyses the system model's data structure provided as UML class model. Additionally, available UIPs are required as input models.

UIP representation. Each UIP is to be modeled statically as a class diagram, which incorporates both presentation and GUI data model elements with appropriate operations. With that representation "the behavioral and structural characteristics of an interaction artifact that provides a solution to an interface design problem" [6] is indented to be modeled. To complement the structure of a UIP, a UML sequence diagram is modeled that describes typical interaction sequences and may include stereotype functions like data loading and change of presentation states. For automation proposes, the sequence diagram can also be expressed via OCL.

UIP selection. During the processing, the UIPs are then matched to recognized structures within the system's class model. In the end, the developer is presented with all possible matching UIPs, which were found suitable for displaying the systems data structures. This may result in multiple choices, but the potential number of applicable UIPs for a given context is reduced to only matching structures.

Limitations. Beale and Bordbar do not claim to have found an ultimate solution. Their UIP representation is not intended to contain detailed pattern descriptions with forces, trade-offs and implementation hints like a full PLML specification would offer. In contrast, they limit their expressed UIPs to certain data structures and selected interaction elements with no user requirements.

Their primary goal was to analyze a system design model or a selected part of it in order to find proper UIPs to display the recognized data structures and to offer an ultimate selection of UIPs. No "aesthetic aspects" [6] or detailed visual design is captured with UIP models. To add these and more implementation related aspects, platform-specific models were suggested named "Device Profile Model" [6] to translate UIP models and generate specific instantiations.

They discuss another issue that stems from system model complexity and its variations, which depend on the skills and likings of the developers. Since a developer may model system design differently, the pattern recognition may produce different results. This is the same issue of varying detail of class diagrams where rather atomic units or composites may be chosen as model elements. Finally, these issues are to be treated by future work and in particular by an enhanced recognition algorithm.

However, the approach by Beale and Bordbar is closely bound to the data structure of a certain context or system. Therefore, the UIP definition is rather narrow and intended to fit within their set limitations. Following this approach, developers will soon seek for a more flexible UIP representation to fit the contexts of task and business process based systems. In addition, no implementation details were given for the data centered UIP concept.

III. Development of Business Information Systems with the Aid of User Interface Patterns

A. *General Graphical User Interface System Development Artifacts*

Before we look into the details of the UIP analysis, we would like to reflect the GUI development process and the potential role of UIPs therein.

General development steps. For each greater business information system the developers have to specify an essential model [31] that captures all necessary functional requirements. The artifacts of this specification are foremost kept independently from architectural and technical details. Therefore, the requirements usually contain no concepts for the GUI system. The transformation of requirements to a final user interface is no easy task to achieve [31]. Several modeling and refinement steps have to be undertaken where means for transformation rarely consist of automation tools. In reference [3], we already explored the theoretical implications of UIPs on these general transformation steps.

Artifact hierarchy. To reflect the role and value of UIPs in these particular development steps, we look closer at the involved requirement artifacts that are displayed on the left hand side of Figure 1. This figure and the following explanations will be used to argue that UIPs may be classified by several types, which reside on considerably different levels in a hierarchy. This UIP structure can be organized in parallel to the architecture artifacts in the middle column of Figure 1. Consequently, the matching UIP types are arranged on the right hand side of Figure 1.

Nowadays, requirements of business information systems are to be structured in a hierarchy of modular artifact types. This is due to the increasing complexity and number of requirements to be implemented. Requirements of higher level organize the structure and referencing of the lower level ones. Redundancies are avoided and concerns that form a modular structure are incorporated. These may lead the software architecture design and help identifying system related or implementation artifacts. For comparable reasons, UIPs should be organized in a similar fashion.

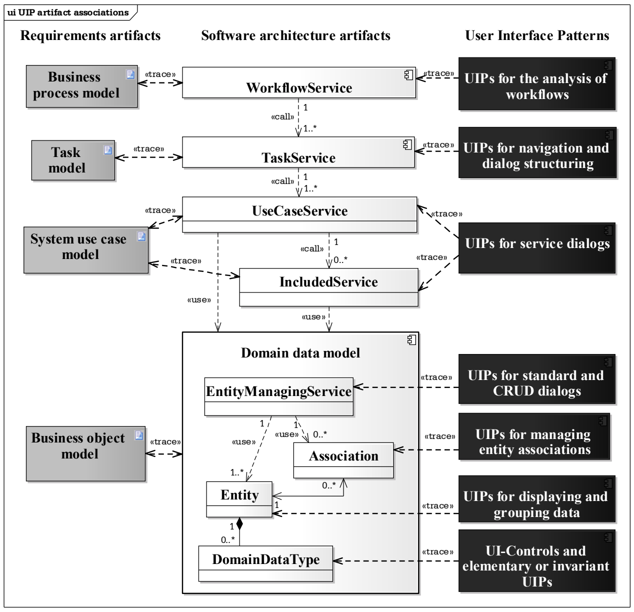


Figure 1. User interface patterns and software architecutre artifact relationships.

They should follow the architecture design levels derived from the requirements structure in order to provide a collection that is modular and reusable without redundancies.

UIPs related to development artifacts. Beginning at the highest level of specification, *business processes* are to be defined as requirements that guide the flow of events and *tasks* from the business goals perspective. They combine the system's as well as the company's resources logically and chronologically in order to realize certain business goals [32]. The part of their specification that is considering the system will be realized via *workflows* and their *services*. The *workflow service* practically is a technical implementation of the IT-supported portions of the *business process*. During its lifecycle it will interact with several applications at once in order to call the individual systems and their GUI implementation that offer access for the user to the realization of requirements situated in lower hierarchy. That is why UIPs will have to be considered mostly for these artifacts. Concerning the workflow itself, there may be UIPs relevant that enable the editing, monitoring and analysis of stored and currently running processes.

The next requirements level in the hierarchy is made up of *tasks*. One can argue whether *tasks* may be settled higher or lower in hierarchy than *use cases*. But that is not our concern at the moment. In Figure 1, *tasks* represent a manual activity of a *business process* as it is perceived by a single user or role. The *task model* captures structure and flow of functions or *use cases* that are combined to achieve the goal of the respective *business process activity*. Thus, the model arranges selected *use cases* to form a flow of events for a certain purpose. As these artifacts are mostly flow oriented, UIPs will be applied here, which determine the navigation and structure of dialog units the user needs to follow. This need was already investigated in reference [33] and acclaimed by other researchers [2][8][9][10] who incorporated respective task patterns.

Situated right beneath the *tasks*, *use cases* (actually system use cases) describe the interaction between user and system on a more detailed level incorporating references to the *domain data model*. In general, the user's goals and the systems provided *services* are specified in this respect. As many interaction steps, events and much data handling may be involved, the UIP type that complements these requirements will provide templates for dialogs that may be adapted to the individual context. Nearly the same applies for *included services* since they are shared among different *use case services*. UIPs may suggest sub-dialog types or portions of them for this type of shared service.

The last level in requirements hierarchy is represented by the *business object model*. The *business objects*, their relationships and data types relevant for higher level requirements are specified herein. Concerning the architecture, Evans [34] suggested an approach that merges analysis and design of respective artifacts into one coherent model, which uses similar stereotypes as depicted in Figure 1 as building blocks. These stereotype classes can be closely associated to certain UIPs. For instance, *entities*, which represent *business objects*, can be displayed by UIPs that arrange their data via tables, forms or other data views. Each time the *entity* is handled, the respective UIP may be instantiated and reused. This also applies to the *association types* the *entities* may use. Specialized dialogs that are applicable for editing certain object *associations* can be abstracted to UIP types. This principle can be followed for standard or CRUD (create, read, delete, update) dialogs, which are used solely for displaying and editing *entity* data. The UIPs only define the similarities of these common and reoccurring dialogs and adapt to the context by parameters like the concrete *entity* or *association* when instantiated. As far as the *DomainDataType* is concerned, there may be only certain UIP types needed for the objects that require a specialized view with a number of interaction options like calendars.

In sum, UIPs may work on different levels of abstraction and may be composed along this hierarchy of their associated artifacts. The requirements and their realizing architecture artifacts use a certain abstraction and structure for good reasons like handling of complexity and avoiding redundancy, so the UIPs should follow a similar structuring for consistent assignment. Finally, the scope for reusable UIPs is vast for business information systems since they should support a set of different architecture artifacts as this is drafted by Figure 1.

User interface development steps. Besides the structuring and assignment to their complementary artifacts, employing UIPs for GUI design involves some more development tasks.

Depending on the level of the considered architecture artifact, several UIPs must be brought together to form the user interface. In this regard, the developer has to arrange for dialog layout, choice and number of UIP instances, UIP instance positioning, and events as well as individual UIP instance visual states definition. Concerning the choice of UIPs, a developer may use the support of any suggestion tool and follow the principle that was presented in reference [6]. Furthermore, the developer needs to integrate the instantiated UIPs with the application kernel and its services. To do so, the UIPs should be able to be configured via parameters for data and action-binding. The former will be required beginning at the lowest level in artifact hierarchy when *DomainDataTypes* are to be bound to single UI-Controls or those contained within UIP instances. With respect to *service* artifacts, UIP configuration must facilitate the binding to actions that trigger the further processing and control by *services* discovered on top of the *domain data model*.

To conclude, there are various structures and related information on each stage to be considered when employing UIPs as reusable pattern artifacts.

B. User Interface Pattern Development and Modeling Concept

In this section, we briefly introduce the general considerations that seem necessary to approach an ideal UIP concept that can be employed in an artifact structure like illustrated by Figure 1.

Domain analysis. A development team may first start with an analysis what parts of the GUI systems are likely to be reused. They can consult existing descriptive UIP libraries like [28][29][30] to gather inspiration for future GUI visual specification. The selection of UIPs may depend on the domain and hierarchy of requirement artifacts.

UIP requirements model. The next step consists of the description of UIP capabilities. Due to the missing general definition of UIPs, there is no consent what are the actual requirements or features that UIPs must fulfill. In the previous section, we argued that UIPs should be sub-divided among several types that reside on a level in hierarchy that matches certain architecture artifacts. The UIPs have to feature properties that allow developers to customize their instances for corresponding artifacts. In addition, reusability and variability have to be specified in detail to enhance configuration facilities. Since UIPs will serve as abstractions for certain parts of the GUI system, they need to enable the same responsibilities with their specification. For all these concerns, an UIP requirements model should be established that fits the intended domain and grade of reuse. In the following section, we will present such a description model for UIPs that has been developed in our previous work.

UIP analysis model. When the requirements or features of UIPs have been pointed out clearly, the development team has to think about what structures, properties and relationships can be derived from the UIP requirements model. The task at hand is about the transformation of those requirements into detailed structures that prepare an information model, which will guide the later formalization of UIPs. This model primarily serves the purpose of a requirements analysis and is not intended for realization. The entities and their relationships derived from the UIP requirements can be modeled via a traditional object-oriented analysis model. As result, the analysis model should express all elements, properties, structures and relationships that will be needed by a language that will be employed to formalize UIPs for automation.

UIP meta model and formalization. On the basis of the analysis model, a formalization concept can be sought of. At this stage, the development team has to decide on the abstraction level the UIP will reside on. More precisely, a decision has to be made how closely UIPs should be bound to target platforms and GUI frameworks. Vanderdonckt [27] presented the Cameleon Reference Framework, which can be consulted for further guidance. In this regard, the abstract user interface (AUI) level groups *tasks* into containers and their structure. Therein, UI-Controls and containers are only defined generically as abstract interaction objects (AIOs). These can be shaped very differently with respect to the next two steps: Concrete user interface (CUI) represents a common platform-independent basis model and final user interface level (FUI) embodies the device or platform model using the specific rendering units of the GUI framework.

We already analyzed this model in reference [3] and came to the conclusion that UIPs should be modeled on the CUI level. The CUI employs concrete interaction objects (CIOs) that refine the AIOs of the AUI. In detail, CIOs resemble a chosen set of both UI-Controls or containers and their respective properties sourced from common UI-toolkits or frameworks. To enable the platform-independent application of UIPs the CUI level should be chosen.

Finally, the developers have to decide on a language that facilitates CUI level modeling of UIPs. Depending on the analysis model, enhancements for existing languages may have to be developed. The parameters and variability concerns most likely need a new concept not already included in languages available in our days. In the end, a metamodel for UIPs has to be established that defines the logical elements being available for the formalization language. The refinement of the UIP metamodel may take several iterations as both analysis and requirements model may be changed several times and gain maturity. Moreover, mandatory and optional elements for UIP formalization have to be determined in order to prepare for different UIP types in the sense of a hierarchy symbolized by Figure 1.

Architecture artifacts metamodel. In parallel to the development of the UIP metamodel, the architecture artifacts of the domain have to be abstracted for forming a separate metamodel. This serves the purpose of mapping UIP types to matching artifact types. The specific artifacts and their stereotype properties have to be determined. The properties will be used to associate potential UIPs to architecture artifacts so that the developers will be presented with choices what UIPs will be generally applicable for a certain context. For instance, a date type as a *DomainDataType* of an *Entity* can be associated to a UIP consisting of a textfield and a connected date selector. Another option could be the presentation of a calendar UIP whenever this *DomainDataType* is encountered. Thus, both metamodels have to establish connections between architecture artifacts and UIPs since architecture properties will partly serve as parameters to enable action- and data-binding when configuring UIP instances.

Transformation concept. After the conceptual modeling has been completed, technical concepts for the transformation of instantiated UIPs into executable dialogs of the GUI system have to be developed. There are several options how to compose a solution. We are still considering these and only mention general directions since they are not in the scope of this work. Concerning principal architectures for generation, reference [3] can be consulted. In addition, there also is the possibility of using interpretation of CUI models. References [10] and [11] briefly described that approach for UIML.

C. *Requirements Model for User Interface Patterns*

Based on our previous work, we progressed towards an elaborate influence factor model for UIPs that is depicted in Figure 2. Motivated by missing standards and competing UIP notations inside modeling frameworks, this model was intended to establish an independent requirements view on the formalization and instantiation of generative UIPs: We took our examples and architecture experiments [3], as well as criteria, aspects and variability concerns [15], and refined them. The requirements stand close to the profile of current approaches in research. For details, reference [5] can be consulted.

As seen in the previous chapters, UIP and architecture artifacts should match. Thus, a UIP definition to be sought after has to introduce a pattern conception, which is backed by a limited set of types, roles, relationships and collaborations among GUI related specifications and components. Because of the complex nature of both GUI architectures and specifications, a restriction and specialization of the entities to be involved in the development environments for pattern-based GUIs have to be set. Along with this restraint, the GUI specific kind of pattern still needs to be abstract in order to enable vast customization and instantiation to differing contexts. The major share of the patterns vigor has to be sourced from the similarity in structural (*view aspect*) and behavioral (*interaction* and *control aspect*) definition of new GUI entities. In other words, the pattern definition introduces certain quality aspects in GUI design, which can be altered quantitatively, when they are respectively complemented with necessary structure, layout and style details (*view variability parameters*) as well as combined with each other (*behavioral* and *structural composition abilities*). This commonality ensures that no longer specialized solutions or manually refined structures, which cannot be covered by mere UIP instantiation, are applied in the same GUI system architecture.

Differences with UIPs. The question may be risen what will be the differences or benefits when taking the efforts to incorporate UIPs in the GUI development process compared to alternatives like GUI builders or CUI level based specification of a user interface with XML languages.

With UIPs, greater units of reuse will be employed as this is the case for CUI level languages and GUI builders. More precisely, complete dialogs or partly views of them can be configured as reusable units. Following the GUI builder or XML CUI specification approach, only small units situated on the UI-Control level or invariant views can be reused.

Along with the reuse of greater units, their interaction facilities and visual states may be reused as well. This kind of reuse is not possible with GUI builders or CUI models. Reuse would only be possible by copying and pasting large portions of existing CUI level code. Subsequently, the code has to be adapted manually to fit the changed context.

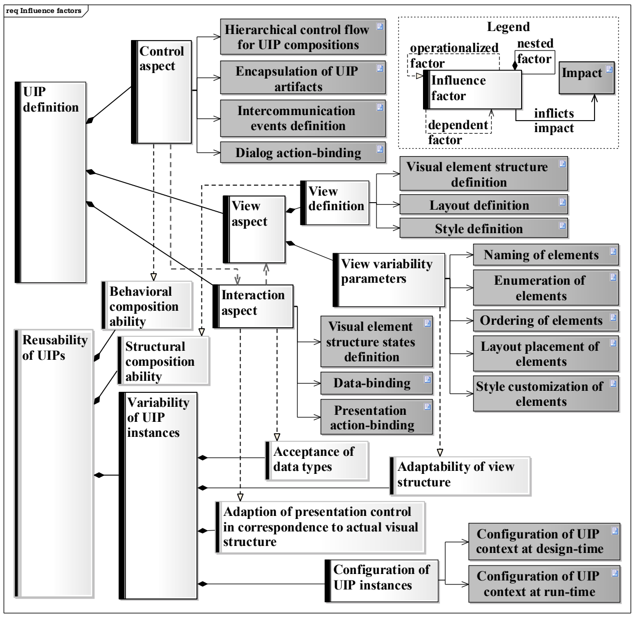


Figure 2. Influence factor model for generative UIPs described in reference [5].

By the application of UIPs, only the declaration of parameters that quantitatively alter the inner structure, states and behavior of defined UIP instances should be necessary in an ideal development environment. The quality aspect, and thus the general structure and behavior, should remain the same for all instances of the same kind of UIP.

Lastly, UIPs may enable their adaptation even at runtime when respective parameters have been specified [5].

Trade-Offs. The main issues while employing UIPs are the high efforts needed to establish a modeling concept or framework as outlined in the previous section. In addition, tools have to be developed that effectively support the developer in the formalization, selection and instantiation as well as rendering of UIPs. Moreover, most parts of the GUI architecture have to be prepared for automated processing with UIPs. In this current work, we only cover one single step of defining the general structure of UIPs via analysis.

IV. Review of Related Work and Problem Statement

A. UIP Definition

Descriptive UIPs. From our observations concerning descriptive UIPs, we learned that they are well-understood as specification elements and supported by the HCI community. Nevertheless, the research into descriptive HCI patterns has not yet converged towards a standardization for the structure and organization of UIPs [17][23].

Generative UIPs. Generative UIPs may be classified as software patterns, and as those, they need a formal notation, and thus, are seldom encountered.

From our point of view, the past work on HCI patterns is concentrated on the descriptive form. As there is no unified approach in specification and usage of descriptive HCI patterns, they can hardly be used to source and abstract common elements of a generative representation. First and foremost, descriptive UIP sources may be a useful resource to assemble dialogs that may act as representative examples for a certain system or domain. On that basis, requirements or criteria for UIP formalization can be inductively obtained. Partly, we revert to this approach and sketch some example UIP instances in Section V.B.

As a consequence, there is a large gap concerning the detailed definition of generative UIPs. Thus, a format for UIPs has to be found that is at least able to express most impacts of *view* and *interaction aspect*. Filling the gap with their own UIP concepts and notations, the UML-based approach by Beale and Bordbar [6] as well as the recent model-based approaches will be analyzed in the following sections.

B. Modeling User Interface Patterns with UML

The approach by Beale and Bordbar introduced in Section II.C directly associates domain data structures to already modeled UIPs. This way a UIP can be derived from the context since the view structure (UIP model) is somewhat similar to the domain model or similarities can be identified thereupon.

Abstraction level. Referring to the Cameleon Reference Framework [27], the UML model of UIPs is situated at AUI level. There are no CUI or any specific visual details mentioned at all. Neither abstract nor concrete UI-Controls to be used for UIP elements are specified. Instead, a final user interface (FUI) level may be generated with the aid of the “Device Profile Model” [6]. It is not entirely clear to what extend the developers have to refine the existing UIP models for their instantiation.

No UIP metamodel. The modeled UIPs and their interaction sequences follow the UML metamodel facilities. There was no specialized UIP metamodel developed. In contrast, each UIP model represents a separate metamodel for certain instances to be created for the FUI. Therefore, they miss a generally applicable UIP description model, which governs adaptation or variability options. Those options are implicitly derived from the domain data model to be supported. The modeled UIP elements will adapt their child elements in correspondence to the attributes provided in the domain data model classes. Thus, the resulting FUI greatly depends on correct and complete modeling of the domain. In this regard, the “overview plus detail” (OPD) pattern might lead to false matches when “item” (C part) may not be detailed enough to justify a full “detail” view. This would depend on the actual “item” data structure and currently is not considered in the OPD UIP metamodel.

UIP factor support. A short comparison with our UIP requirements model reveals that certain aspects cannot be covered by the UML approach. Only *view* and *interaction* aspects are partly covered.

The *control* aspect or pattern composition is not directly considered at all. The structural UIP composition ability may be implicitly included when greater parts of a domain model or more classes with a number of relationships are analyzed. Then either multiple UIPs will be suggested to be applied together or a greater UIP metamodel has to be incorporated that matches the complete structure. Nevertheless, overlapping pattern definitions or composite UIPs that do already employ smaller UIPs are not addressed with the required attention.

There are more restrictions concerning the *interaction* aspect. The partitioning or querying of data may not be prepared. In detail, the domain data must be displayed or processed on the GUI as modeled in the application kernel, so certain views or queries that may alter or merge data structures cannot be used for the original UIP selection. Otherwise the GUI data model must be specified separately and in detail so that the pattern recognition may finally work.

Many variability or parameter related impacts are to be derived implicitly from the domain. This applies for naming, ordering layout and style specification of UIP instance elements. Data- and action-binding may only be adapted in fixed limits of the defined sequence diagrams or OCL specification. The developers cannot configure non data-intensive aspects such as navigation, dialog structure and preparation of data inside views or dialogs and their level of detail with the reuse of available UIPs. Sometimes only selected attributes of an entity may be needed for display and not the entire attribute set of a domain class.

Benefits. Apart from these limitations, the aim of Beale and Bordbar primarily was to reduce the amount of UIPs to be taken into consideration for a certain *domain data model*. It may be beneficial when UIPs suitable for a certain *task* are to be suggested on the basis of a complex data structure available for analysis.

Supported artifacts. However, this may be a great restriction since the types of employable UIPs will be limited to certain levels inside the *domain data model* of Figure 1. So far, the UML approach only supports certain levels and special artifact relationships. UIPs are not subdivided concerning artifact support. In the next section, model-based approaches based on modeling frameworks mostly offer a more subtle classification of pattern types or their structures.

C. Summary of Model-Based Development Processes involving User Interface Patterns

The enhancement of model-based development by generative UIPs already found strong reception. In reference [5], we presented an overview and assessment of the approaches of Zhao et al. [2], PIM [35], UsiPXML [8], PaMGIS [9] and Seissler et al. [10]. For a summary, Table I compares these approaches.

In sum, the model-based approaches are converging concerning the *view* aspect, but ultimately failed to convey or inspire all UIP impacts. A summary of realized (arrow in a box) or inspired (single arrow) impacts is given by Figure 3.

Since our valuation revealed that there were many open issues associated with the different approaches, we only considered the full and no partly or probable realization of an impact. Notably is that the *view aspect* was realized by the most recent approaches. In contrast, the *interaction aspect* was only considered for *Data-binding*. Moreover, the *control aspect* was not realized by any approach, but inspired by PIM. Lastly, the *Configuration of UIP instances* was restricted to design-time only, but already inspired by Seissler et al. in reference [11].

TABLE I. COMPARISON OF APPROACHES FOR MODEL-BASED DEVELOPMENT EMPLOYING USER INTERFACE PATTERNS

	Approach			
	Zhao et al.	***UsiPXML***	***PaMGIS***	***Seissler et al.***
Pattern types	Task patterns based on [28], set of window and dialog navigation types	Task, dialog, layout and presentation	Task and presentation patterns, fine grained hierarchy based on	Task, dialog and presentation patterns
UIP formal-ization notation	Unknown	Enhanced UsiXML	Unknown, XML based, <automation> tag and DTD	Embedded UIML supplemented by parameter and XSLT enhancements
UIP config-uration	At design	At design	At design	At design and run-time
Process output	Target code	UsiXML, M6C	Target code	Augmented UIML to be interpreted

Concerning the architecture artifacts, the approaches already incorporated pattern types dedicated to certain abstractions in the hierarchy of their modeling framework. Thus, the idea of matching UIPs and architectural artifacts or even patterns inspired by Figure 1 is already incorporated in those approaches to some extent. However, as they lack a clear requirements and structural definition of UIPs the mapping between artifacts cannot be considered as fully elaborated.

	Arrangement of elements within defined layout	Configuration of UIP context at design-time	Configuration of UIP context at run-time	Data-binding	Dialog-action binding	Encapsulation of UIP artifacts	Enumeration of elements	Hierarchical control flow for UIP compositions	Identification and distinction of UIP categories	Intercommunication events definition	Layout definition	Naming of elements	Ordering of elements	Presentation action-binding	Style customization of elements	Style definition	Visual element structure definition	Visual element structure states definition
PaMGIS	$\boxed{\uparrow}$						$\boxed{\uparrow}$				$\boxed{\uparrow}$	$\boxed{\uparrow}$	$\boxed{\uparrow}$				$\boxed{\uparrow}$	
PIM					↑	↑		↑		↑								
Seissler et al.	$\boxed{\uparrow}$	$\boxed{\uparrow}$	↑				$\boxed{\uparrow}$				$\boxed{\uparrow}$	$\boxed{\uparrow}$	$\boxed{\uparrow}$			$\boxed{\uparrow}$	$\boxed{\uparrow}$	
UsiPXML	$\boxed{\uparrow}$	$\boxed{\uparrow}$		$\boxed{\uparrow}$			$\boxed{\uparrow}$				$\boxed{\uparrow}$	$\boxed{\uparrow}$	$\boxed{\uparrow}$				$\boxed{\uparrow}$	
Zhao et al.				$\boxed{\uparrow}$														

Figure 3. Impacts covered by examined approaches.

D. Formal GUI Languages and Model-Based Development

Enhancements. As there is still no dedicated language for UIP formalization, developers have to revert to existing GUI specification languages like UIML or UsiXML, which enable the specification of GUI parts on the CUI level. We will refer to them as XML languages in the following. As a result, two factions among the model-based approaches arose, one using UsiXML and the other applying UIML. Both languages need enhancements to express UIP related variability. Accordingly, the model-based approaches incorporated their own parameter and configuration concepts. In sum, they all failed to publish enhancements that empower the specification languages regarding the *interaction* and *control* aspects. Currently, the notations are restricted to the *view* aspect mostly.

Generation of XML specifications. The XML languages have been developed to offer a platform-independent specification of GUI systems. In this context, they have been based on a metamodel that is somewhat similar to common universal object-oriented programming languages, which cannot handle aspects or traits and thus are incapable of expressing patterns with their abstract form. The XML languages clearly fail in the fulfillment of the reusability, variability and composition ability criteria [3][15].

However, applying the XML languages for their original purpose, apart from pattern definition, may play out their strengths. Accordingly, developers could use them for concrete GUI definition and final rendering to the desired platform. To integrate UIPs in this procedure, a generation of XML language code could be a possible solution to overcome the inabilities as proposed in reference [3]. This idea was already followed either by generation of UsiXML [8] or the interpretation of UIML [10]. The XML code would hold the already instantiated UIPs or the required information for rendering. The benefit would be the possibility to use existing tools for the XML languages. In addition, a more important merit would exist in obtaining a concrete user interface level (CUI) specification [27], and thus, the ability to be independent from platform specifics. In sum, the UIPs and their instantiation would be used to create CUI level models either based on UIML or UsiXML. The CUI model could be processed by the tools and transformed to target platform FUIs.

In any case, a new language or extensions for the XML languages are to be sought after. Whether UIPs are being defined concretely in XML or the latter is generated, the XML languages will surely be a fundamental part of this solution. Consequently, the new language must facilitate the expression of UIP instances in rich XML language specifications. For that purpose, a unified UIP-model has to be established, which truly holds all information for the definition of generative UIPs and parameters for their transformation to UIP instances or instance compositions forming a concrete GUI model on CUI level.

V. Our Approach

A. *Strategy*

As mentioned in the objectives, the impacts in reference [5] resulted in the strategy to develop an analysis model, which is aimed at further detailing the UIP aspects. We develop a structural model that is biased towards an implementation of a dedicated UIP language.

Motivation of an analysis model. Some requirements such as *interaction* and *control aspects* are cross-cutting concerns and are really hard to achieve for pattern formalization. Thus, more planning and rationale is required before we can consider the development of a dedicated language. We follow the way of traditional modeling of requirements and ease their transformation to design with an analysis model. The model is intended to express the domain terms and concepts with a structure.

With a structural and more detailed model, the tracing of the influence factor impacts to potential solutions is better possible than with the pure influence factor model presented by Figure 2. In the factor model, there exist no separated entities that are modeled with their attributes and relationships to reflect a possible solution approach.

Assessment of recent approaches. Although we pointed out the factor support and issues we could so far discover as result of our assessment of other available approaches in reference [5], we also concluded that more details on examples and the applied notation have to be revealed in order to refine the assessment. By developing an analysis model, we seek to overcome the lack of detail and rationale on the design of notations suitable for UIPs. The notation to be used for modeling is the UML 2.0 class model.

Why do we propose a semi-formal model? For a technical architecture design or a generative process for formal UIPs to be verified, a wide range of requirements emerging from the initial criteria have to be taken into account, which cannot comprehensively modeled on a formal basis. In contrast to other researchers directly pushing towards a formalization of UIPs, we think this intermediate step is necessary and helpful. In our opinion, a semi-formal model is more useful to the developer than a formal model in first place, hence the mental conception about full scale generative UIPs has to be inspired first. The understanding of these complex patterns, their aspects and element relationships is the primary goal that should not be hindered by formal media, which cannot be imagined easily. A semi-formal model enables a better understanding than a grammar, since it may visualize concepts, their structure and relations depending on the chosen notation.

In sum, the model has to satisfy the information needs of the developers first, before they can think of how to employ the available formalization options or even GUI XML languages to express the requirements residing inside the model. Primarily, the model has to capture requirements in way that is easily understandable for human-beings.

Why do we apply a UML 2.0 class model? The UML class model lies in between the descriptive nature of the factor impacts and a formal notation. In this regard, a class model is already inclined towards a formal implementation. This is the case for class models serving as a design model for object oriented programming languages. In analogy, our analysis model may lead to a design for new language elements for the definition of generative UIPs. The language to be sought after also should rely on a structural paradigm, since the GUI implementations form a structure as well.

Moreover, a class model already proved useful for the expression of design patterns. The paradigm employed allows us to model abstract data types, their common attributes as well as their cardinalities and relationships. As the model entities all reside on an abstract level and do not describe already instantiated objects, the class model proves to be suitable for our task. More precisely, the UIP concepts can be modeled from a point of view where the abstraction and instantiation are separated. The class model forces the developer to express his solutions by abstractions that concentrate the commonalities of later instantiated objects. As we seek to express UIPs that feature reusable GUI solution aspects, a class model may provide a proper notation.

With the class model, we will be probing the modeling of required information for UIPs. Currently, developing a particular language or focusing on a certain architecture experiment seems to be too specific. In contrast, we investigate how the information of UIPs and their configuration can be established in general. To sort out possible options, trace factor impacts on more detailed granularity and map them to the final solution, the analysis class model may prove as a valuable asset. Finally, we may draft a coupling between a UIP, its configuration and GUI architecture or at least mandatory prerequisites.

B. User Interface Pattern Examples

By reason that we do not want to claim being able to establish a UIP analysis model applicable for each domain, we stick to business information systems as mentioned in the introduction. More precisely, as stated in Section IV.A, we rely on common dialogs for e-commerce applications as a basis. In fact, we subsequently derive the analysis model by focusing both on the factor model in Figure 2 and the following example dialogs.

Simple search. For an easy example, we start with a dialog that has the “Search Box” [28] pattern instantiated. The simple search illustrated in Figure 4 is mainly composed by a single panel (*ContentPanel*), which defines a GridBagLayout as seen in the upper part of Figure 4. The UI-Controls are fixed and aligned in respective fashion. For variability, only the concrete object data types need to be bound to the combobox and textfield. In fact, this kind of UIP is mainly invariant.

Advanced search. The next example shall be more complicated and thus, demand for every aspect described within the factor model. We decided for an “Advanced Search” [28] pattern, which alters its visuals and interaction options depending on user input.

Our example, depicted in Figure 5, mainly consists of two panels for layout definition as shown on the upper half. The panel *RootPanel* defines a GridBagLayout consisting of three cells (grey borders). Located in the center of this container, the *SearchCriteriaPanel* defines a layout of several rows each containing on cell (solid black borders). Additionally, the latter may grow or shrink in height to accommodate or discard search criteria lines to fit inside the container. Lastly, the *SearchCriterionPanel* (dashed borders) defines a layout appropriate for individual search criterions.

The usage of this dialog is as follows: Firstly, the user selects an object to be searched from the “Type of Object” combobox. Secondly, he chooses an attribute from the combobox inside the *SearchCriteriaPanel*.

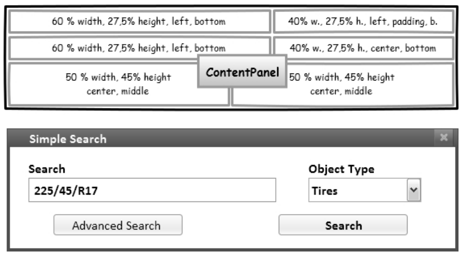


Figure 4. Simple search UIP example layout and dialog.

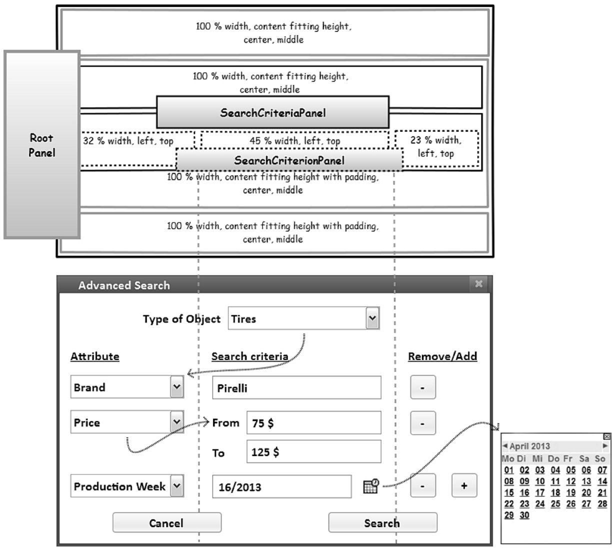


Figure 5. Advanced search UIP example layout and dialog.

Accordingly, the UIP dynamically has to instantiate new sub-UIPs, which resemble the single search criteria rows. For each datatype, a pre-defined UIP, which is similar in shape to the *SearchCriterionPanel*, is assumed to be available. In the example, the datatypes String, price, and week are considered. With the buttons on the right hand side, the user may add or drop new search criteria rows and so the view aspect will change. The variability is limited to the object types and their attributes to be searched with this UIP. Controller related aspects have to be adapted based on the UIP definition.

VI. The Analysis Model

In this section, we develop the proposed analysis model. At first, we review each UIP aspect and its associated impacts in order to elaborate the decisions in design of the new model. Afterwards, we present the structure of the model and finally apply the model to both examples introduced in Section V.B. The terms in *italics* refer to respective analysis model elements.

A. Analysis Model Bias

On principle, there are two options on how to bias the model. Firstly, the model could be biased towards the software architecture and thus employ proven design patterns in its structures. This option would be rather suitable for generators and the further automated processing of the model, but it would be tedious to translate it back to the UIP requirements for the developers. In addition, the formal XML GUI languages (Section II.B) were not designed to accommodate architectural knowledge.

Secondly, the analysis model may be biased towards requirements and thus acting as a traditional analysis model, which captures, refines and visualizes requirements. This option would be rather easy for the developers to understand, but would be costly to be translated to formal languages and generators. However, the translation to the XML languages is only a theoretical aspect, since generative UIPs cannot be expressed by their facilities as discussed in Section IV.D. Eventually, we decided for the latter option.

B. General Rationale

Separation of definition and instances. A fundamental decision was the separation of elements or features that may be available in a UIP definition and the several element instances that may appear in a particular UIP application for a certain context. In other words, we divided the UIP analysis model into two parts. One part holds the definition and reoccurring elements (class names in black). The other part allows the description of instance information (class names in white) and individual element configurations. These basics are illustrated by Figure 6.

UIP configuration. Following the general concept of Figure 6, the main class *UserInterfacePattern* takes part in relationships that mostly focus on definition purposes, but also is connected to *UIPConfiguration*, which enables the description of particular UIP instances of the respective kind. The information used for pattern definition purposes will be covered in the following sub-sections. The configuration of UIP instances further branches into *Defaults* and *Parameters*. Both classes resemble containers that hold the *UIControl* instance information, which is declared as *UIControlConfigurations*, for a particular UIP instance. The *Defaults* are intended to omit stereotype configurations of default *UIControl* instances, which commonly appear in most contexts and shall not be re-defined redundantly.

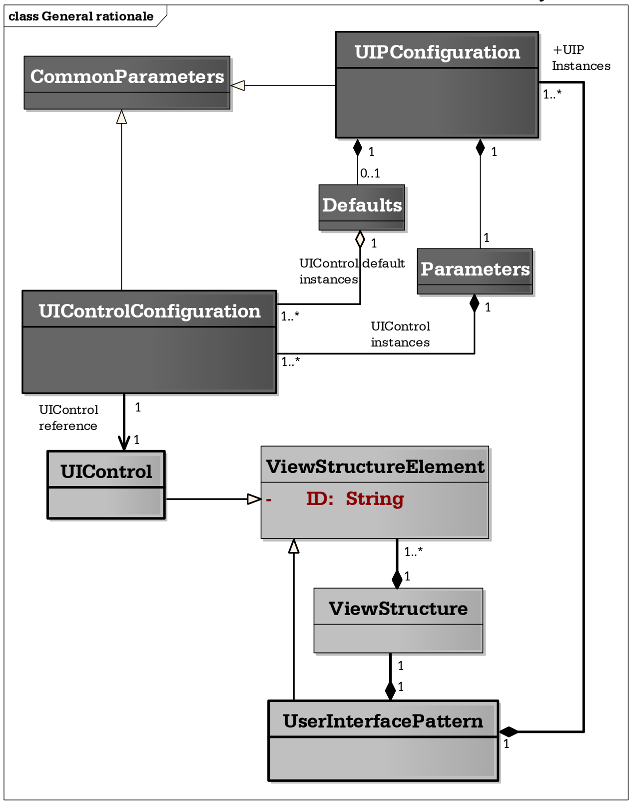


Figure 6. Gerneral rationale of the UIP analysis model.

Concerning the example dialogs, the basic or invariant *UIControls* needed for user understanding and interaction like the labels, textfield and combobox of the simple search should be defined as *Defaults*, as there is hardly any variability. This way, already established configurations may partly be reused among individual UIP instances. That means a UIP may contain pre-configured elements and parameters to avoid repetition. Later on, this facility will become useful for the dynamic adaptation of a UIP instance at run-time. Both *UIPConfiguration* and *UIControlConfiguration* are primarily used for the "Configuration at design-time" impact and thus contain the declarations a developer may define in interaction with an "instantiation wizard" [8] or any other configuration tool.

The configuration of *UserInterfacePatterns* and *UIControls* has to be separated, since both offer different sets of attributes, and more important, impact the GUI on different levels of abstraction or scope. This consideration also takes the possible artifact relationships of Figure 1 into account.

C. View Aspect Design

View definition. To begin with "View definition", this factor defines the *UIControls* or *UserInterfacePatterns* to be generally contained or allowed in a UIP specification unit as visual components. Both resemble a *ViewStructureElement*, which has a unique *ID* as identifier inside the pattern used by *UIPConfiguration* and *UIControlConfiguration* to reference the respective element. In this respect, *UIControl* is a classifier for the various visual components or widgets a GUI framework may possess as types. Figure 7 details the described relationships.

A UIP is always composed of a *ViewStructureElement* set, and thus, may build a varying hierarchical structure of those graphical elements. However, *ViewStructure* only holds each *ViewStructureElement* to be available to build instances once. The resulting element structure of a particular UIP instance is not described by *ViewStructure*. Instead, this is the responsibility of the configuration classes. In other words, from the available *ViewStructureElements* the developer may create instances using the respective configuration facilities. The *ViewStructure* only defines what elements are generally available for the particular UIP. Based on that decision, the *ViewStructureElements* later may be exchanged without altering the already defined configurations. For each *UIControl* of the resulting *ViewStructure*, style and general layout have to be defined.

The style impact is not detailed here, since we have not came to a result in this regard and focused on the other impacts. For the sake of uniform views and maintaining corporate design, style information may be governed globally and locally by each individual *UIPConfiguration*. In addition, there may be constraints for each element, which determine its allowed minimum and maximum occurrences.

Layout rationale. With respect to "Layout definition" impact, we ask if there is a need for dedicated layout-patterns or if the distinction between primitives (*UIControl*) and composites (*UserInterfacePattern*) is adequate.

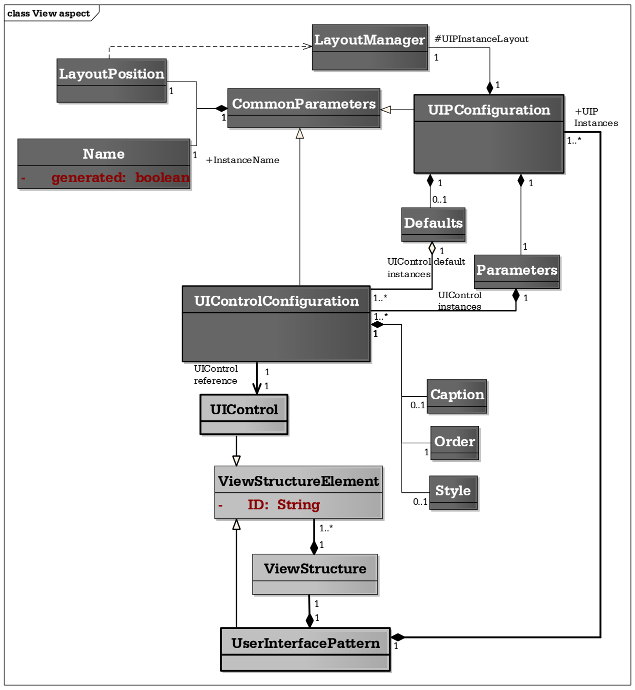


Figure 7. View aspect concepts of the UIP analysis model.

Referring to UsiPXML [8], layout patterns can be defined separately from presentation patterns. How they are integrated at various stages in the hierarchy, and more important, how they can be handled dynamically at run-time, remains an open issue, as there were no detailed examples for pattern composition and specification code given.

In addition, it is arguable whether a layout is assigned separately to a paralleled UIP composition or if each UIP models layout partly but explicitly. Partly means that UIPs need to define attributes for the number of rows and columns of a grid, their relative width and height, as well as the alignment. A visual impression of the abstract layout definition expressed by UIPs is depicted in the upper parts of Figure 4 and Figure 5. We decided to model this information by UIPs, as for advanced search, the layout needs to be re-configured dynamically with respect to *SearchCriteriaPanel*. This panel may grow and shrink in row numbers.

Layout definition. Inspired by our examples, we treat the layout container as a UIP, and thus, a layout pattern is already merged inside. So, the above mentioned layout definition parameters have to be associated to each *ID* of a UIP-type class, since it is acting as a superior container. Consequently, the advanced search dialog consists of three UIPs designated as containers in Figure 5. Translated to GUI frameworks, this implicates that each UIP will be treated as a panel or even window frame with a certain *LayoutManager* attached. We reason our approach with the fact that every dialog at some stage needs layout containers and these are eventually to be mapped to peers in the GUI framework. The detailed parameters for layout, such as padding, orientation and size policies, may be governed globally.

View variability parameters. To configure parameters for an element of the *ViewStructure*, regardless of what type, the respective *ID* of that element is used as a reference.

The *UIControlConfigurations* assigned to UIPs influence the instantiated unit in a global way. So, for the *view aspect* the general layout of the instances *ViewStructure* is declared by *LayoutManager*, which decides on the actual grid, for example. This way, the layout and orientation of UIP instances may be altered, but have to be declared explicitly for each *UIPConfiguration*.

As the elements defined by a UIP are abstract, the reference to the *ID* acts in analogy to the class concept for object-orientation. In fact, the element occurrence is determined by the number of respective configurations. For the individual element instances, one or many *UIControlConfigurations* can be declared to specify their characteristics. More precisely, as *view aspect* parameters we arranged for *Name*, *Caption*, and *Order* inside a layout grid cell and *Style* of each element. Some of these parameters are even optional. With *LayoutPosition*, the position of the element with respect to the declared *LayoutManager* can be defined.

Both *UserInterfacePattern* and *UIControl* share some parameters defined as *CommonParameters*. For both *ViewStrucutreElements* the *Name* and *LayoutPosition* may be declared.

D. *Basic Interaction Aspect Design*

In the factor model of Figure 2, the *interaction aspect* was not separated between stereotype definitions and parameters, as this was done for *view aspect*. Finally, the main classes, which model the *interaction aspect*, all do resemble parameter types. Since the factors apart from the *view aspect* ones mostly embody cross-cutting concerns, the resulting *interaction* and *control impacts* refer to the static and variable declaration of *view impact* elements as a basis. This relationship is outlined with the dependency between *view* and *interaction aspect* in Figure 2. In detail, the *interaction* related *UIControlConfiguration* parameters comprise of *DataType*, *PresentationEvent* and *EventContext* as an additional child of the latter.

Coupling points. For a UIP definition to be integrated in a GUI architecture, there is the need to arrange for coupling points. These points allow the integration of automated generated code and manually defined UIP information. Potentially, these can be comprised of the following:

- Standard events (*control* - "intercommunication events definition", "dialog action-binding")
- Input and output data (*interaction* - "data binding")

The latter point may resemble GUI architecture models discovered in common MVC architectures. The mentioned coupling points are either evaluated (events) or processed by the dialog kernel or logic part of the dialog. It is not necessary for that component to know where data changes and events have originated from. So, these suggested coupling points may be a good starting point. Accordingly, events (*PresentationEvents* and *OutputActions*) and the "GUI Data Model" have been included in the analysis model. These features originate from our thoughts about artifact relationships in Section III.A, and in particular, the association of *domain data model* elements and UIPs.

Data-binding. The binding of a *UIControl* to certain data is accomplished by a *UIControlConfiguration* parameter. So, the *DataType* binds the elements to certain data structures. As *DomainDataTypes* may significantly differ from the types used by the GUI framework, the class *GUIProjection* is rather associated as the configured *DataType*. For the *DataType*, it can be configured if the data is to be displayed only (input) or if the user may conduct changes (output), which are finally applied to the GUI *Model* part. The *DataType* parameter also may be associated to *EventContext*, which configures the data to be submitted by a *PresentationEvent* of the respective element. The diagram of Figure 8 details the *interaction aspect* data-binding considerations.

Besides the distinction between input and output, *Models* have to be provided as coupling points for both cases to obtain data for display. The application kernel has to provide a respective query to obtain *Entity* data and the GUI architecture has to implement a certain *Model* to enable the presentation of the query with appropriate data types for *UIControls*, e.g., data conversion to strings or string lists. In this regard, aspects like the timing, refresh rate, lazy loading are no concern of the UIP definition and have to be implemented by the data sources or queries. The *Model* has to rely on the data source and is not responsible of those technical aspects. In contrast, the *Model* needs to provide the navigation inside data structures and the structuring of data for presentation purposes that may be altered from application and data layer designs in order to offer a suitable projection for human processing.

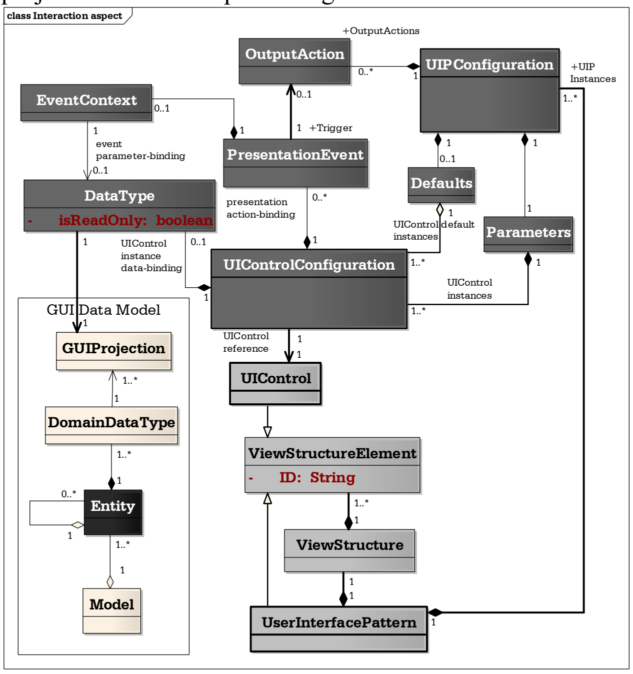


Figure 8. Interaction aspect data-binding concepts of the UIP analysis model.

Currently, we are unsure how UIPs specific *Model* requirements are to be formalized. However, this information is essential for the coupling. In addition, it will prove useful for the checking of the validity of configuration and *view* variability of the UIP instance. Concerning the advanced search, there must be a *Model* available to provide object types and their attributes as well as another *Model* to accommodate the chosen search criteria as the dialog result.

Events rationale. For *PresentationEvents*, we enumerated some typical events implemented in GUI frameworks that may be triggered. To progress towards a unified solution for generative UIPs, we think that a standardization of events, *PresentationEvent* as well as *OutputAction*, and similar types is necessary. Figure 9 displays elements of the analysis model relevant for events.

The integrative and strict type definitions of the GUI specification language UsiXML on CUI level [27] may be a valuable resource for that approach. Otherwise, both specification and tool processing would demand for niche solutions that are hardly manageable with respect to versions and dependencies. We wonder how UsiPXML [8] or the UIML UIP definition by Seissler et al. [10] are defined as a language to be integrated in tool environments, which are to handle the generic concept of their variables and assignments effectively. We have to wait for them to publish detailed language definitions and code examples.

Presentation action-binding. To bind an element to a certain *PresentationEvent* type, the desired event has to be included in the appropriate *UIControlConfiguration*. This event may be declared for various purposes concerning view structure states as described below.

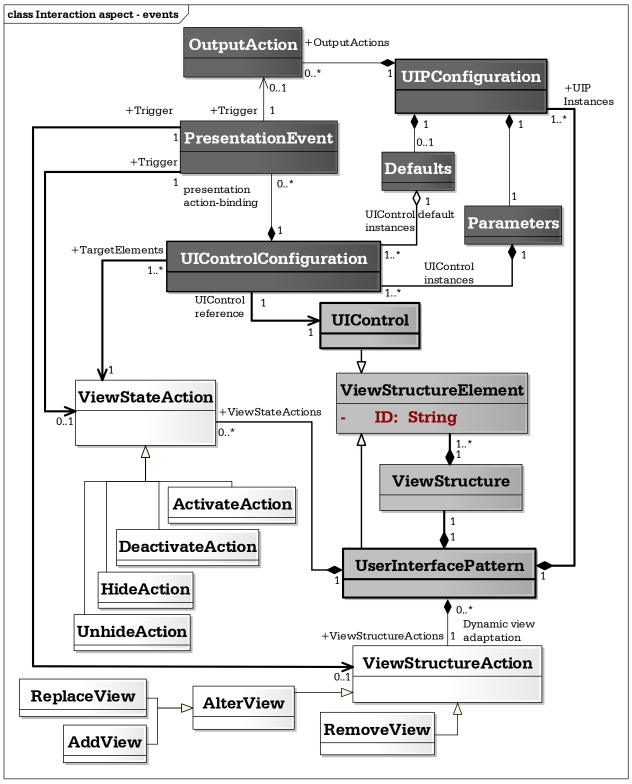


Figure 9. Interaction aspect event concepts of the UIP analysis model.

E. General Object Model View

To clarify the basic rationale the UIP analysis model is founded on, we will explain the general structure of an UIP artifact viewed from an object model's point of view. Figure 10 illustrates the basic objects to appear in each UIP specification. The structure of the analysis model leads to a hierarchical ordering of elements to be used for UIP specification.

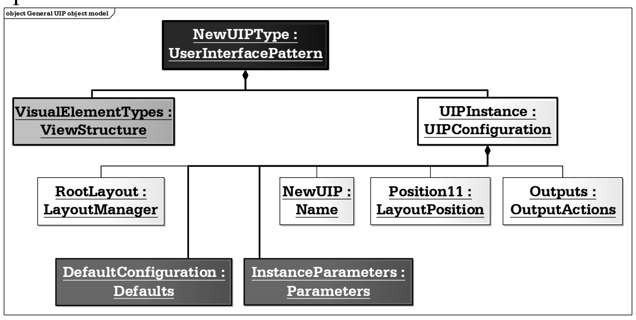


Figure 10. General structure of a UIP artifact based on the analysis model.

In this regard, the first two objects to appear are one *ViewStructure* for the definition of available *UIControls* or even nested *UserInterfacePatterns* and one *UIPConfiguration* that will be used to adapt the UIP instance object to the context. A *Name* and *LayoutPosition* with respect to the parent *LayoutManager* are to be specified as *CommonParameters*.

The parameters are shared among *UserInterfacePattern* and *UIControl* objects, so that both kinds of *ViewStrucutreElements* may be named and placed concerning layout.

One *UIPConfiguration* object with a reference to the main *UserInterfacePattern* object is mandatory. There may be more than one *UIPConfiguration* object when nested *UserInterfacePatterns* do occur within the *ViewStructure* object. With the respective *UIPInstance* object, all instances based on the available elements from the *ViewStructure* will be created. In addition, the general layout or *RootLayout* and the *OutputActions* relevant for architecture integration will be defined with that object, too.

The *UIPInstance* object holds two more configuration objects. On the one hand, a *Defaults* object will enable the reuse of the common configuration of that particular UIP. Therein, stereotype instances created from the available *UIControl* elements of the *ViewStructure* will be configured for the convenience of reuse. On the other hand, context-specific instances based on the *ViewStructure* specifc *UIControl* elements can be created with the *Parameters* object in parallel.

F. Advanced Interaction Aspect Design

Visual element structure states definition. The first *interaction aspect* impact needs to be further detailed. Depending on the actual structure of the UIP, states that occur within the scope of the contained *UIControls* and states, which alter the view of embedded UIPs have to be covered. To trigger changes in state for both cases, only *UIControls* can be specified as sender of respective events.

***UIControl* states.** For changes in state, we consider the activation or deactivation as well as hiding and un-hiding of single *UIControls* or sets of them. Those abstract events are to be translated to technical representations and their detailed implementation. For instance, a checkbox in a sub-form may deactivate the delivery address (if it is equivalent to billing address) or in another case, a collapsible panel may be collapsed. In our model, the *ViewStateAction* is defined as an abstract feature for a UIP. By the UIP specification, the possible actions are defined and associated to affected *UIControlConfigurations*, and thus, *UIControl* instances. Finally, triggering *PresentationEvents* can be associated for these actions.

Embedded UIP states. Since the possible states for composite UIPs cannot be enumerated or state machines finitely defined inside pattern specifications, we employ information, which describes the results of the state change, and thus, enables a generator to build appropriate state machines or comparative implementations. In Figure 11, relevant elements for the specification of dynamic UIP structures are displayed.

The *ViewStructureAction* is designed to handle the change of visual states for UIPs. For the trigger, a respective *UIControlConfiguration* is needed, which is aimed at a certain *ID* to allocate the *UIControl* and the type of *PresentationEvent*. We considered the addition, replacement, or removal of UIP instances. This behavior is closely related to the <restructure> tag of UIML [36] and may be refined based on its semantics. However, for UIML these facilities can only be applied with already instantiated UIPs.

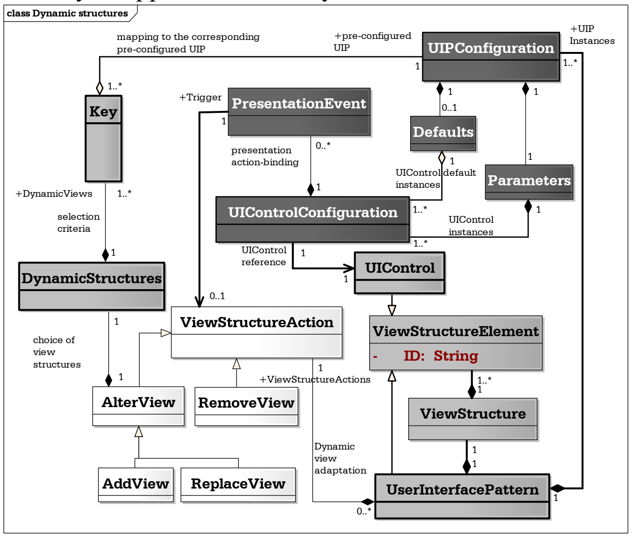


Figure 11. Interaction aspect embedded UIP states concepts of the UIP analysis model.

DynamicStructures are used for the addition, removal or replacement of *UserInterfacePattern* instances at runtime. They are selected on the basis of defined *Keys*, which enumerate certain *DataTypes* or *EventContext* data to assign pre-configured *UIPConfigurations* to the triggered *ViewStructureAction*. A *UIPConfiguration* may be used by more than one *Key*, which models a certain context situation. Concerning the advanced Search example, the *Model* holding the object and attributes lists must return values that match the specified keys. Each time a combobox is changed, the presentation event handling routine must query the *Model* for the selected object's attribute and its kind or type of representation. The query result will be embedded in the *EventContext*, which is matched to a *Key* value. This way, the UIP and its *DynamicStructures* are based on a canonical representation of *DomainDataTypes*.

Moreover, the *ViewStructureActions* rely on pre-configured elements, which may only allow for variability concerning the *DataType*. They either rely on a self-reference (removal, replace) or additionally are associated to available elements of the *ViewStructure* (add, replace) via *DynamicStructures*. However, this mechanism only makes sense for *UserInterfacePatterns*, which are specified by *Defaults* and always represented by default *IDs* present inside the *ViewStructure* of a UIP definition. In this way, the *DynamicStructures* will only affect default or invariant *UserInterfacePatterns* inside the given *ViewStructure*, hence it is not desirable to replace entire sets of UIP instances defined on behalf of the developer for a specific context. Thus, manually defined UIPs portions have to be separated from *DynamicStructures*.

Based on the considerations for *DynamicStructures*, we decided to associate *DataType* with *GUIProjection* rather than with *DomainDataType*. A reference to *DomainDataTypes* would have meant to define a *Key* and appropriate *UIPConfigurations* for each *DomainDataType*. Each change of types would have cascaded to each UIP relying on *DynamicStructures*. We believe that *GUIProjections* may be more stable than *DomainDataTypes* and even be shared among *DomainDataTypes*.

G. Control Aspect Design

Dialog action-binding. So far, we have not progressed to feasible results for most *control aspects*. Only the binding of *UIControls* to application actions has been included. Via the global *OutputAction* parameter declaration of a UIP, one can define what events of that kind are raised by the *UIControlConfigurations*. These can be bound to a certain *UIControl* only by a link with the *PresentationEvent*.

H. Structure View on the Analysis Model

The resulting analysis model is illustrated by Figure 12. The classes shaded in medium grey are related to the "view definition" factor. Configuration related classes are shaded in dark grey and feature a white caption. Most *interaction aspect* impacts are supported by the classes shaded in white.

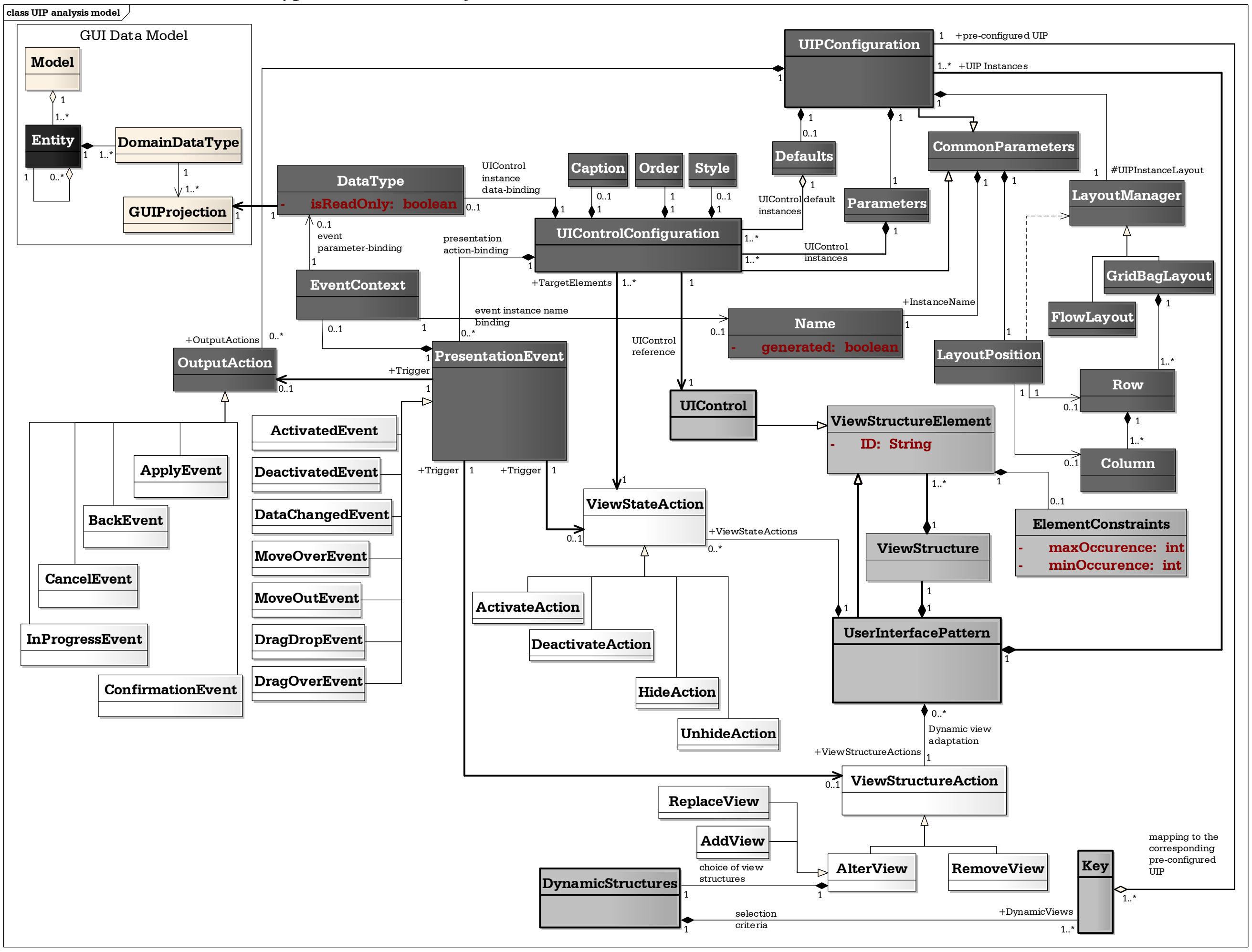


Figure 12. User interface pattern analysis model.

VII. Instance View on the User Interface Pattern Analysis Model

A. Simple Search Instance View

In this section, we apply the above described analysis model to the first UIP example entitled simple search. For that purpose, object models will be presented that are used to illustrate the different aspects of the UIP instance configuration. Please note that due to space limitations not all mandatory associations or objects will be modeled.

Since the simple search is mostly an invariant UIP, there is a need for a default configuration. Instance parameters will be limited to the *DataTypes* associated to the search input textfield and objects to be searched, which are determined by the user through the combobox listing available object types.

ViewStructure. To begin our analysis of that example, we enumerate the *UIControls* that are to appear as visual elements in the *ViewStructure* of the UIP. There are labels for designating the visual elements for the user, a texfield for search input, a combobox that holds object data and two buttons for triggering *OutputActions*. Each of these elements was incorporated into the *ViewStructure* on the left hand side of Figure 13. The label and buttons only appear once since their needed instances will be configured as *Defaults* holding *UIControlConfigurations* accordingly.

Defaults. The tree with *DefaultConfiguration* models the real UI-Controls that will appear on the screen when simple search is instantiated. For each *UIControl*, the caption and layout position are specified. Some labels have been skipped for presentation purposes; their configuration is performed in analogy to the other labels present in the object model. As far as the buttons are concerned, additional *PresentationEvents* are declared that are of the type *ActivatedEvent*.

UIControlConfiguration. Besides *Defaults*, the *UIPConfiguration* declares a *LayoutManger* used for positioning the *UIControl* instances. The two possible *OutputActions* are also specified on the same level.

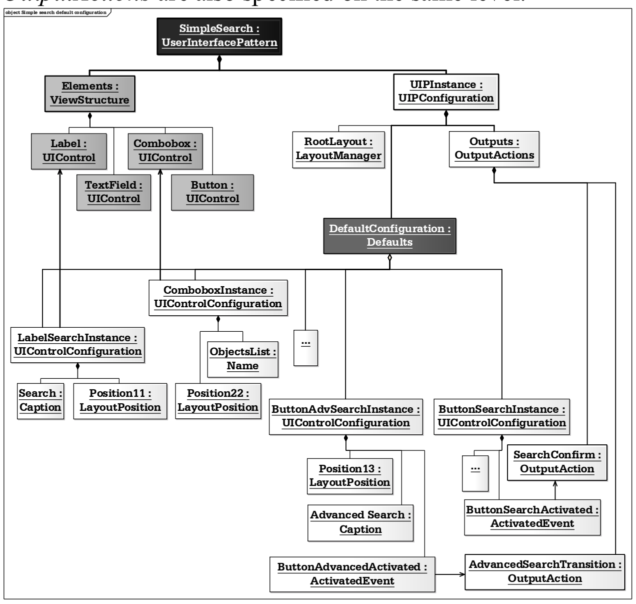


Figure 13. Simple search instance default configuration object model.

Lower in hierarchy, they are associated to triggering *PresentationEvents* that belong to certain *UIControl* instances, and more precisely, their *UIControlConfigurations*. This information is typical for that kind of UIP and can be reused by the *Defaults*.

Variability parameters. To adapt the UIP instance to the specific needs of the context, *Parameters* will be declared as depicted in Figure 14. The missing information for the data relevant *UIControls* is added here. To reference the same *UIControl* instance, the same *Name* has to be used during specification. For instance, the textfield and combobox are already present inside the *Defaults* object.

For both the texfield and combobox the data-binding is specified with reference to an existing *GUI data model* based on the *Entity SearchObjectData*. The latter will be processed by an application kernel service and has no GUI related responsibilities.

To increase the variability of that UIP, one could think about adapting the layout via parameters but this is currently not reflected in our analysis model.

The example is rather simple since the UIP has no visual states that cannot be handled implicitly by the facilities of the implementing GUI framework. The complete object model is provided with Figure 15.

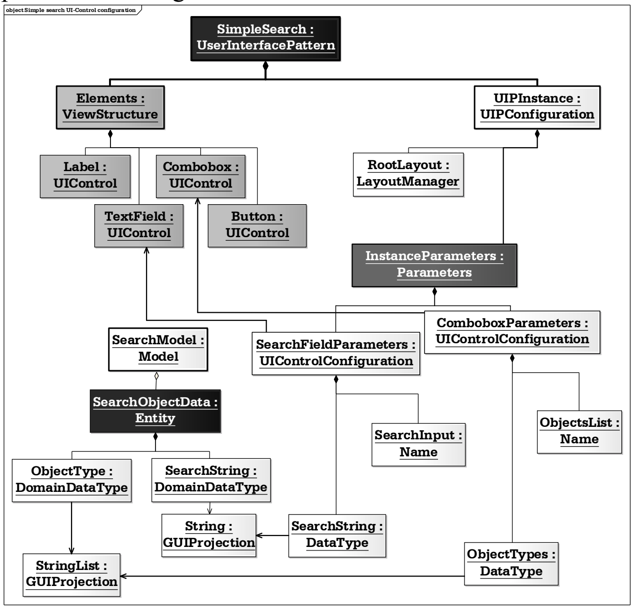


Figure 14. Simple search instance UI-Control configuration object model.

B. Advanced Search Instance View

The advanced search is far more complicated than the simple search object model. Therefore, we begin with a state chart that displays a set of a few possible alternations of the view state. In Figure 16, the state chart of the advanced search example is illustrated.

It is obvious that the visual element structure states are altered, each time the user performs a relevant input such as the selection of an object, an attribute or the activation of a button. The advanced search example involves a complex structure of nested UIP instances.

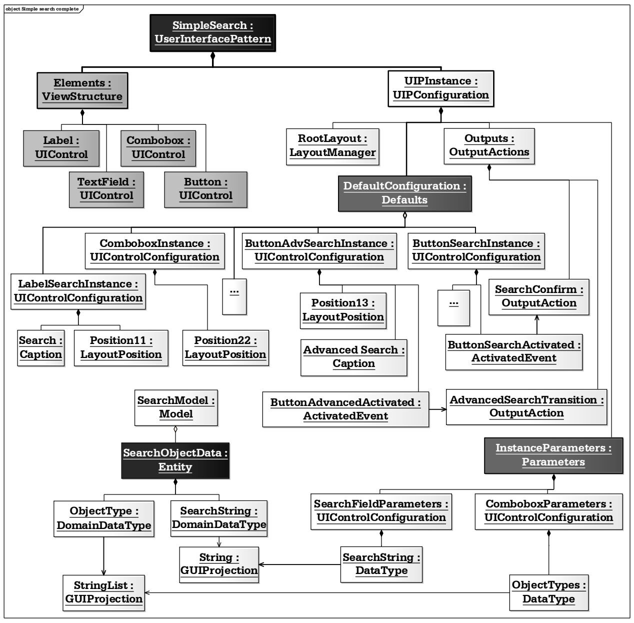


Figure 15. Simple search instance complete object model.

To begin with, the main instance will be one object of the advanced search UIP itself. Moreover, the pattern consists of a lower *ButtonBar* and a *SearchCriteriaPanel* that is built dynamically during user interaction. These basic UIP instance objects are arranged in the object diagram of Figure 17. The *MainInstance* holds configuration information about the data origin of the object types, so that the user may begin with a selection of the object to be searched. When the user has made his selection, a *DataChangedEvent* will be triggered as a consequence of the user interacting with the first combobox of the dialog and its respective *UIControlConfiguration*.

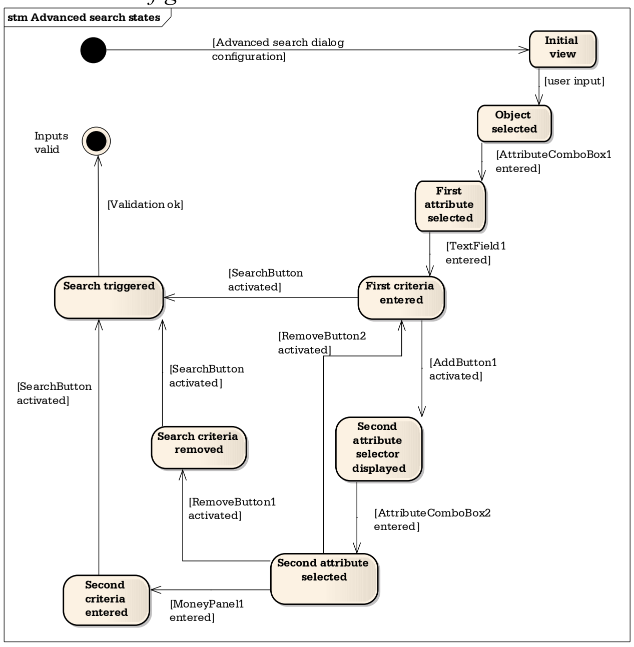


Figure 16. A few selected states of the advanced search example UIP.

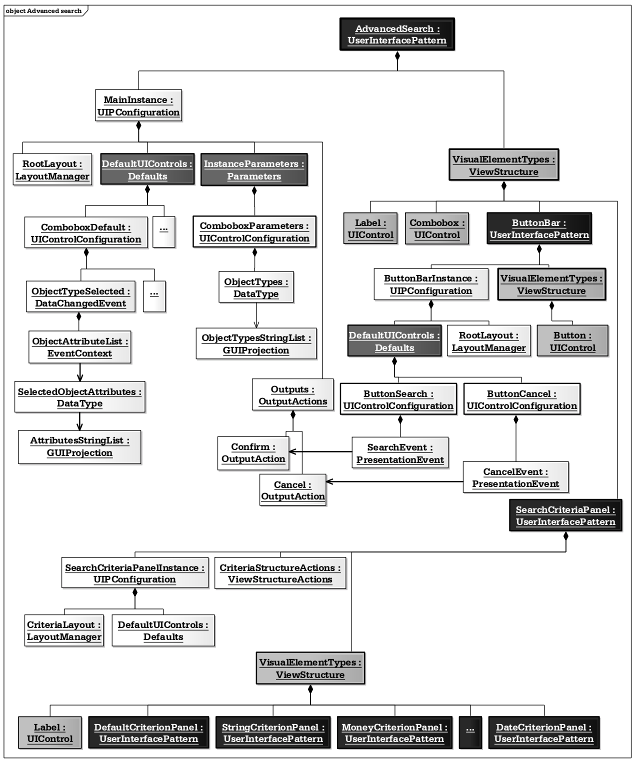


Figure 17. Advanced search basic object diagram.

The event will be associated with an *EventContext* to submit the attribute list of the selected object type to the lower situated search criteria comboboxes. The *SearchCriteriaPanel* constitutes of a number of embedded *UserInterfacePatterns*. These UIPs will serve as templates for the dynamic instantiation of search criterions. Notably is the *DefaultCriterionPanel*, which will be instantiated first when a selection has not been made by the user yet.

A detailed view on the *SerachCriteriaPanel* reveals the structure of the embedded *DefaultCriterionPanel* that is available in the *ViewStructure* of the former. The *DefaultCriterionPanel* defines a *ViewStructureAction* that allows for the replacement of the complete UIP instance with a pre-configured *UserInterfacePattern* of the parent *ViewStructure*. This replacement will be triggered when the attribute to be searched is entered by the user. A respective combobox *UIControlConfiguration* is present in the *DefaultCriterionPanel* default configuration. Depending on the attribute selected, an appropriate *UIPConfiguration* is determined via the evaluation of the *EventContext* and stored *Keys* of the *DynamicStructures*. Figure 18 provides a detailed object diagram.

Finally, with Figure 19 a partly object model of the advanced search example is presented.

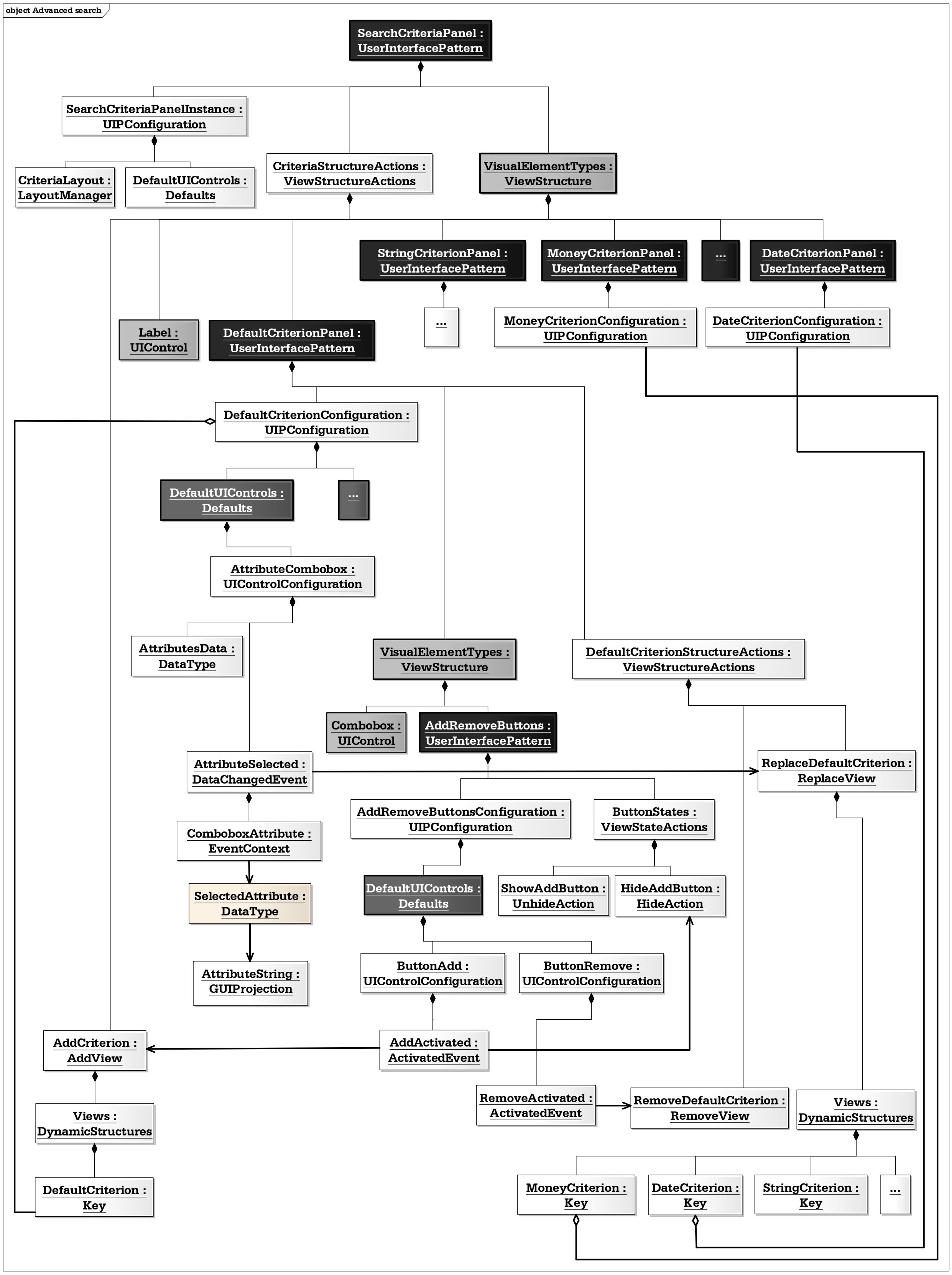


Figure 18. Advanced search example *SearchCriteriaPanel* object model.

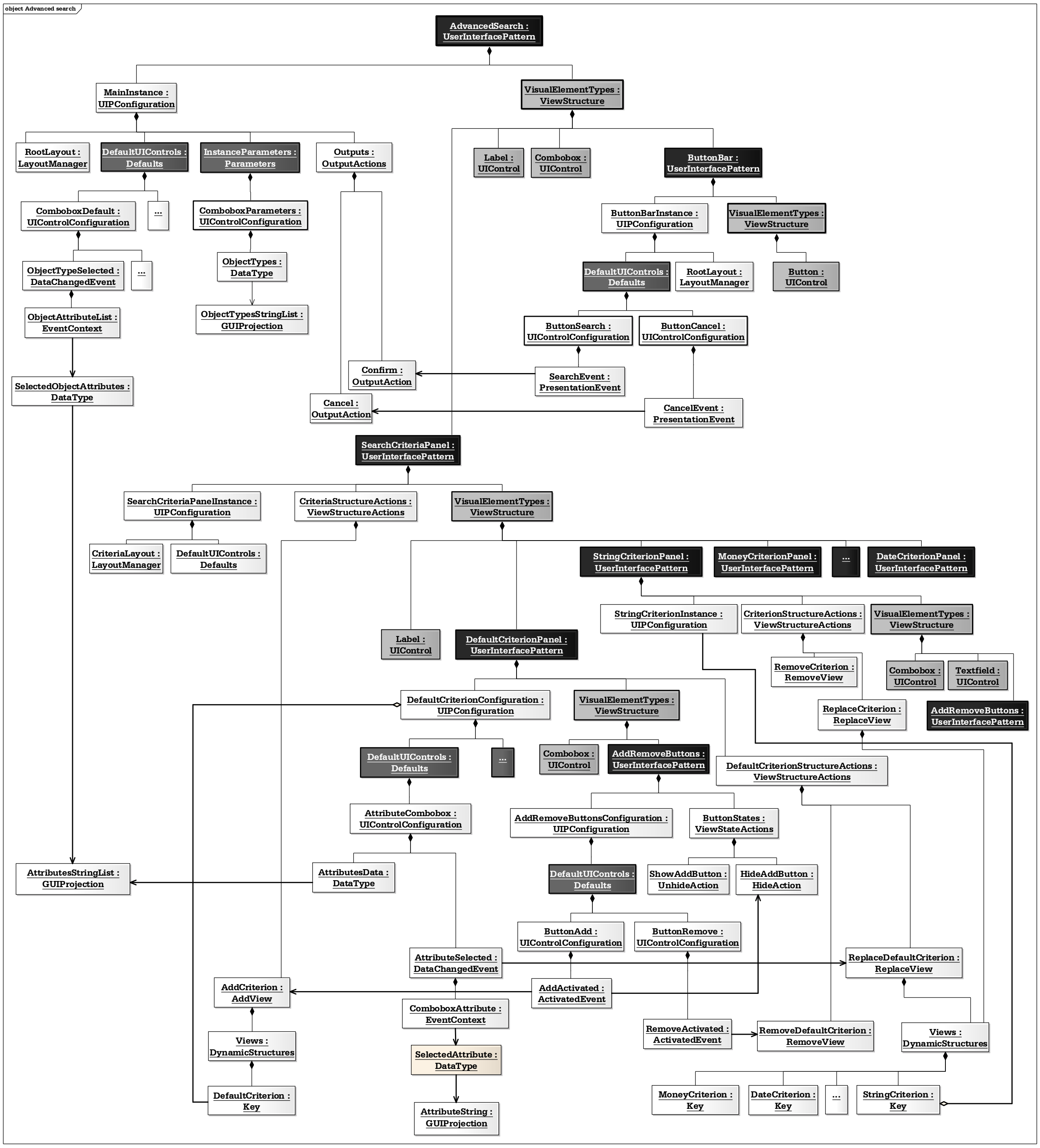


Figure 19. Advanced Search partly object model.

VIII. Results and Discussion

Achievements. With the elaboration of our analysis model, we detailed most factor impacts of our previous work on requirements for generative UIP representations [5][16]. Accordingly, we proposed fine-grained structures, which are in closer proximity to real applicable pattern notations than pure requirements can be.

Judgment. The current state of the analysis model is quite imperfect. However, with this initial iteration we achieved a better understanding of the information needed to express UIPs and their instances. A more vivid impression on requirements, which we have modeled explicitly and are implicitly supported by current approaches employing UIPs for model-based development [5], has been gathered. Furthermore, the model already may be used to verify the capabilities of notations for generative UIPs.

The object models gave us a good impression on the current state of the analysis model. Furthermore, the probing of the expression of selected example UIPs has proven a quite a good coverage of needed description elements. Much of the required information already could be modeled. However, the advanced search example is quite complex and could not be described here in more detail.

Moreover, the further analysis of the object models will reveal what impacts are yet to be enhanced. Not before the analysis model has been improved and object models can satisfactorily be expressed, we can think about a possible formalization of UIPs. This approach avoids unnecessary iterations and trial & error during the design of a dedicated UIP language.

The potential notation, generator tool-chain and especially the generated architecture, which may be derived in the future from the analysis model, most likely will be somewhat complex, but since they are solely intended for automated processing without manual interference, this is a trade-off for a step further to implement generative UIPs.

Again, we would like to invite other researchers to contribute either critical judgments or improvements for the presented analysis model or its requirements basis.

Traceabilty. Concerning the realized impacts, we established traceability-links between the analysis model classes and the factor impacts of our requirements model. Figure 20 displays the relationship matrix accordingly. Please note that only generalized classes are included in the matrix. This is due to the fact that specialized classes do inherit the links from their parent classes.

As result, the analysis model almost fully complies with the elaborated requirements. Currently, “Hierarchical control flow for UIP compositions” and “Intercommunication events definition” do remain unsolved from the *control aspect* impacts. As far as the *view aspect* is concerned, the “Style definition” is an open issue.

Therefore, our analysis model has gained maturity and its elements may serve for the verification of modeling frameworks for generative UIPs on the presentation level. Thereby, the applied concepts, tool in- and outputs and especially the facilities of UIP formalization notation can be traced to the elements of the analysis model.

	Configuration of UIP context at design-time	Configuration of UIP context at run-time	Data-binding	Dialog action-binding	Encapsulation of UIP artifacts	Enumeration of elements	Hierarchical control flow for UIP compositions	Intercommunication events definition	Layout definition	Layout placement of elements	Naming of elements	Ordering of elements	Presentation action-binding	Style customization of elements	Style definition	Visual element structure definition	Visual element structure states definition
Caption											↑						
CommonParameters	↑																
DataType			↑														
Defaults	↑																
DynamicStructures		↑															
ElementConstraints																↑	
EventContext			↑														
Key		↑															
LayoutManager									↑								
LayoutPosition										↑							
Name											↑						
Order												↑					
OutputAction				↑													
Parameters	↑																
PresentationEvent													↑				
Style														↑			
UIControl																↑	
UIControlConfiguration	↑					↑											
UIPConfiguration	↑	↑			↑	↑											
UserInterfacePattern																↑	
ViewStateAction																	↑
ViewStructure					↑											↑	
ViewStructureAction		↑															
ViewStructureElement					↑											↑	

Figure 20. Traceability matrix: Factor impacts realized by analysis model classes.

Unsolved control impacts. Currently, our model only supports *ViewStructures*, which consist of UIPs always being in close cooperation. Nested UIPs are not yet intended to be reused outside the specification or their super-ordinate UIP. Being aware of this barrier, we may need to define facilities such as pattern interfaces, as this was proposed by both UsiPXML [8] and Seissler et al. [10]. In this regard, the *OutputAction* may be refined to accommodate the events required for UIP inter-communication. Eventually, the *UIPConfiguration* may be supplemented by certain input types. In the end, the first three *control aspect* impacts remain unsolved for now.

Open issues. We are aware that our model needs further elaboration and especially verification. Further issues to be solved persist in the classification and delimitation of UIP specification units. The relationships among UIPs discussed by Engel, Herdin and Märtin [23] may be considered, too.

IX. Conclusion and Future Work

Ultimately, we drafted an analysis model for UIPs by resuming our previous work on requirements towards a definition for generative UIPs. As result, the analysis model already covers mandatory structures and expresses variability aspects of generative UIPs. Together with our factor model, the analysis model may be taken into consideration for the verification of the capabilities of other UIP based approaches or languages mentioned and not mentioned here. Our object model examples proved that current elements of the analysis model are required and sound for certain UIP instances.

However, the presented user interface pattern analysis model has to be reviewed and refined with the aid of other researchers in the future in order to establish a focused basis for a sophisticated generative pattern definition. With the latter, a dedicated notation can be developed that will allow the modeling of a vast and flexible range of UIPs. On that basis, a more thoroughly applicable solution concerning the covered GUI parts will be available in theory. In the long run, maybe the complete GUI system can be expressed with generative UIPs and their customized instances, even if this means to capture system specific or custom UIPs in the same specification format for the sake of a unified generative process. As time has revealed, several model-based GUI generation solutions and processes have emerged and came to their limits when the need for fully variable UIP instances came up. For generative UIPs, the path to a mature definition has been paved by our contributions.

In contrast, design patterns have already evolved over time and found a common notation and expression. This was feasible because the shape and aspects for that kind of patterns concentrated on the general abstraction of object-oriented paradigm, which is easier to grasp for developers as its reach in system responsibilities is very general and universal, and most important, limited to the concepts of a reduced set of repeating classes or their object instances. In addition, design patterns are applicable for many domains, and with reference to Figure 1, vast parts of their respective architecture artifact levels. For comparison, UIP modeling concepts tend to be bound to certain artifact levels governed by specific modeling frameworks. Thus, their UIP definitions are not flexible to allow for a combination with artifacts on varying levels. Simply put, the UIP pattern concepts have not reached a vast and general applicability as design patterns did.

Concerning the GUI architecture and patterns, there is no single shared definition or interpretation for MVC that can be relied upon. The pattern functions as a mental model layer on object-orientation for developers to classify the complex responsibilities and flows of GUIs by using atomic universal components customized for higher architecture understanding. Looking at our proposed analysis model, instantiable UIPs with their variable aspects in this abstract diagram are considerably more detailed and complex compared the initial MVC representations.

To conclude, we have to strive for a better understanding of UIPs, and after some iterations, a common UIP concept resembling the maturity of design pattern will be established finally. In the end, we have to enhance the available work on model based GUI development processes with the new UIP definition. Alternatives do not exist, as the presently available solutions offer no common generative UIP definition and thus can only cover a small portion of the GUI system, do not allow for sufficient variability or architecture artifact coverage. Again, we do need to strive for a common generative UIP definition in order to derive a conceptual basis the technical implementations and tools can be based on.

Future work. For future work, we see a refining and correcting iteration for the analysis model with regard to simplicity and completeness according to all impacts. In detail, we have to assess the mandatory and optional parameters on the basis of our listed examples. Furthermore, we will concentrate on the unsolved control aspect issues. With the progression towards an improved version of our analysis model, a more general applicable model-based UIP development process may be established in the future.

After completion of the UIP analysis and definition, we plan to consider options how to establish a notation that will be a realization of our analysis model. As candidates for GUI specification languages, UIML and UsiXML are likely to be considered for basic foundations. Eventually, for both languages enhancements will have to be developed in order to enable the support of generative UIPs.

REFERENCES

[1] S. Wendler and D. Streitferdt, “An analysis model for generative user interface patterns,” The Fifth International Conferences on Pervasive Patterns and Applications (PATTERNS 13) IARIA, May 27 - June 1 2013, Xpert Publishing Services, pp. 73-82, ISSN: 2308-3557.

[2] X. Zhao, Y. Zou, J. Hawkins, and B. Madapusi, “A business-process-driven approach for generating e-commerce user interfaces,” The Tenth International Conference on Model Driven Engineering Languages and Systems (MoDELS 07), 2007, Springer LNCS 4735, pp. 256-270.

[3] S. Wendler, D. Ammon, T. Kikova, I. Philippow, and D. Streitferdt, “Theoretical and practical implications of user interface patterns applied for the development of graphical user interfaces,” International Journal on Advances in Software, vol. 6, nr. 1 & 2, pp. 25-44, 2013, IARIA, ISSN: 1942-2628, http://www.iariajournals.org/software/.

[4] G. Meixner, F. Paterno, J. Vanderdonckt, “Past, present, and future of model-based user interface development,” i-com, vol. 10, issue 3, November 2011, pp. 2-11.

[5] S. Wendler, D. Ammon, I. Philippow, and D. Streitferdt “A factor model capturing requirements for generative user interface patterns,” The Fifth International Conferences on Pervasive Patterns and Applications (PATTERNS 13) IARIA, May 27 - June 1 2013, Xpert Publishing Services, pp. 34-43, ISSN: 2308-3557.

[6] R. Beale and B. Bordbar, “Pattern tool support to guide interface design,” Human-Computer Interaction - INTERACT 2011 - 13th IFIP TC 13 International Conference, Part II, Sept. 5-9 2011, Springer LNCS 6947, pp. 359-375.

[7] A. Wolff, P. Forbrig, A. Dittmar, and D. Reichart, “Tool support for an evolutionary design process using patterns,” Workshop on Multi-channel Adaptive Context-sensitive Systems (MAC 06), May 2006, pp. 71-80.

[8] F. Radeke and P. Forbrig, “Patterns in task-based modeling of user interfaces,” The Sixth International Workshop on Task Models and Diagrams for Users Interface Design (TAMODIA 07), Nov. 2007, Springer LNCS 4849, pp. 184-197.

[9] J. Engel and C. Märtin, “PaMGIS: A framework for pattern-based modeling and generation of interactive systems,” The Thirteenth International Conference on Human-Computer Interaction (HCII 09), Part I, July 2009, Springer LNCS 5610, pp. 826-835.

[10] M. Seissler, K. Breiner, and G. Meixner, “Towards pattern-driven engineering of run-time adaptive user interfaces for smart production environments,” The Fourteenth International Conference on Human-Computer Interaction (HCII 11), Part I, July 2011, Springer LNCS 6761, pp. 299-308.

[11] K. Breiner, G. Meixner, D. Rombach, M. Seissler, and D. Zühlke, “Efficient generation of ambient intelligent user

interfaces," The Fifteenth International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 11), Sept. 2011, Springer LNCS 6884, pp. 136-145.

[12] M. J. Mahemoff and L. J. Johnston, "Pattern languages for usability: an investigation of alternative approaches," The Third Asian Pacific Computer and Human Interaction Conference (APCHI 98), 1998, IEEE Computer Society, pp. 25-31.

[13] A. Dearden and J. Finlay, "Pattern languages in HCI; A critical review," Human-Computer Interaction, vol. 21, issue 1, special issue: Foundations of design in HCI, 2006, pp. 49-102, http://www.tandfonline.com/doi/abs/10.1207/.U4hygHe8C4I.

[14] J. Borchers, "A pattern approach to interaction design," Conference on Designing Interactive Systems (DIS 00), August 17-19 2000, ACM Press, pp. 369-378.

[15] D. Ammon, S. Wendler, T. Kikova, and I. Philippow, "Specification of formalized software patterns for the development of user interfaces," The Seventh International Conference on Software Engineering Advances (ICSEA 12) IARIA, Nov. 2012, Xpert Publishing Services, pp. 296-303, ISBN: 978-1-61208-230-1.

[16] S. Wendler and I. Philippow, "Requirements for a definition of generative user interface patterns," The Fifteenth International Conference on Human-Computer Interaction (HCII 13), Part I, July 2013, Springer LNCS 8004, pp. 510-520.

[17] K. Breiner, M. Seissler, G. Meixner, P. Forbrig, A. Seffah, and K. Klöckner, "PEICS: Towards HCI patterns into engineering of interactive systems," The First International Workshop on Pattern-Driven Engineering of Interactive Computing Systems (PEICS 10), June 2010, ACM, pp. 1-3.

[18] M. van Welie, G. C. van der Veer, and A. Eliëns, "Patterns as tools for user interface design," in Tools for working with guidelines, C. Farenc and J. Vanderdonckt, Eds. London: Springer, pp. 313-324, 2000.

[19] J. Tidwell, Designing Interfaces. Patterns for Effective Interaction Design. Beijing: O'Reilly, 2006.

[20] E. Hennipman, E. Oppelaar, and G. Veer, "Pattern languages as tool for discount usability engineering," The Fifteenth International Workshop Interactive Systems. Design, Specification, and Verification (DSV-IS 08), 16-18 July 2008, Springer LNCS 5136, pp. 108-120.

[21] S. Fincher, PLML: Pattern language markup language. http://www.cs.kent.ac.uk/people/staff/saf/patterns/plml.html, 2014.05.30

[22] S. Fincher, J. Finlay, S. Greene, L. Jones, P. Matchen, J. Thomas, and P. J. Molina, "Perspectives on HCI patterns: concepts and tools (introducing PLML)," Extended Abstracts of the 2003 Conference on Human Factors in Computing Systems (CHI 2003), ACM, 2003, pp. 1044-1045.

[23] J. Engel, C. Herdin, and C. Märtin, "Exploiting HCI pattern collections for user interface generation," The Fourth International Conferences on Pervasive Patterns and Applications (PATTERNS 12) IARIA, July 2012, Xpert Publishing Services, pp. 36-44, ISBN: 978-1-61208-221-9.

[24] J. Vanderdonckt and F. M. Simarro, "Generative pattern-based design of user interfaces," The First International Workshop on Pattern-Driven Engineering of Interactive Computing Systems (PEICS 10), June 2010, ACM, pp. 12-19.

[25] M. Abrams, C. Phanouriou, A. L. Batongbacal, S. M. Williams, and J. E. Shuster, "UIML: An appliance-independent XML user interface language," Computer Networks, vol. 31, issue 11-16, Proceedings of WWW8, 17 May 1999, pp. 1695-1708.

[26] Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, and V. López-Jaquero, "USIXML: A language supporting multi-path development of user interfaces," in Engineering human computer interaction and interactive systems, Joint Working Conferences EHCI-DSVIS 2004, July 11-13 2004, Revised Selected Papers, R. Bastide, P. A. Palanque, and J. Roth, Eds. Heidelberg: Springer LNCS 3425, pp. 200-220, 2005.

[27] J. Vanderdonckt, "A MDA-compliant environment for developing user interfaces of information systems," The Seventeenth International Conference on Advanced Information Systems Engineering (CAiSE 2005), June 13-17 2005, Springer LNCS 3520, pp. 16-31.

[28] M. van Welie, A pattern library for interaction design. http://www.welie.com, 2014.05.30.

[29] Open UI Pattern Library. http://patternry.com/patterns/. 2014.05.30.

[30] A. Toxboe, User Interface Design Pattern Library. http://www.ui-patterns.com, 2014.05.30.

[31] M. Ludolph, "Model-based user interface design: Successive transformations of a task/object model," in User interface design: Bridging the gap from user requirements to design, L. E. Wood, Ed. Boca Raton, FL: CRC Press, 1998, pp. 81-108.

[32] H. Umbach and P. Metz, "Use Cases vs. Geschäftsprozesse. Das Requirements Engineering als Gewinner klarer Abgrenzung," Informatik Spektrum, vol. 29, issue 6, pp. 424-432, December 2006, Springer, ISSN: 0170-6012.

[33] A. Wolff and P. Forbrig, "Deriving user interfaces from task models," Workshop Model Driven Development of Advanced User Interfaces (MDDAUI 09), Feb. 2009, CEUR Workshop Proc. Vol-439.

[34] E. Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston, MA: Addison-Wesley, 2004.

[35] F. Radeke, P. Forbrig, A. Seffah, and D. Sinnig, "PIM tool: Support for pattern-driven and model-based UI development," The Fifth International Workshop on Task Models and Diagrams for Users Interface Design (TAMODIA 06), Oct. 2006, Springer LNCS 4385, pp. 82-96.

[36] UIML 4.0 specification, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uiml. 2014.05.30.

Dynamic Suppression of Sensory Detail Saves Energy

Terry Bossomaier*, Lionel Barnett†, Michael Harré,‡ and Herbert F. Jelinek*
*Centre for Research in Complex Systems
Charles Sturt University
Bathurst, Australia
Email: {tbossomaier,hjelinek}@csu.edu.au
†Sackler Centre for Consciousness Science
School of Informatics
University of Sussex
Brighton, BN1 9QH, UK
Email: lionelb@sussex.ac.uk
‡Complex Systems Group
Faculty of Engineering and Information Technologies
The University of Sydney
Sydney, Australia
Email: mike.harre@gmail.com

***Abstract*—High functioning autistic people can exhibit exceptional skills with numbers, eidetic imagery and recall of concrete detail, as brought to popular attention in the film Rain Man. However, it now transpires that these skills are to some extent latent within all of us. We do not have access under normal circumstances to this concrete detail, because it is dynamically inhibited by higher level concepts. Brain stimulation using Trans Cranial Magnetic Stimulation or Direct Current Stimulation, which blocks this inhibition, releases savant-like skills in non-savants. This paper proposes that one of the reasons for this lies in the brain's need to conserve energy. Computer simulations using a spiking neural network support this hypothesis. A spiking neural network was set up with a number of feature detectors feeding an output unit, which in turn generates inhibition of the input neurons. This reduces the spike activity of the input, and thus overall energy usage. We introduce a theoretical analysis for the gains, which might be made. Thus, we demonstrate that energy conservation is a possible cause of inhibition of sensory detail by high level concepts.**

***Keywords*—energy, spiking neuron, inhibition, simulation, sensory detail.**

I. INTRODUCTION

The evidence from high functioning autistic individuals shows the overwhelming advantage of concept formation in the human brain. Such individuals tend to have weak concept formation but can have very powerful perception and memory for detail. The evolutionary significance is abundantly clear. What is less clear is *why the raw detail, to which these people have access is not available to everybody else.* The surprising thing, revealed by direct brain stimulation, is that this detail is *not* destroyed on the way to conscious awareness, but is somehow *blocked from access.* This paper provides a novel solution to this conundrum, which was first described in Bossomaier et al. [1].

To demonstrate that such a conjecture is feasible, we ran a series of computer simulations of spiking neurons and found, that, indeed, energy could be saved in this way. Access to raw detail could create a better brain, able to accomplish a greater range of tasks and arguably be more creative [14]. Thus, a powerful argument is needed for why such a brain has not evolved. Energy could be one such factor.

The energy cost of neural computation is split between the generation of spikes and synaptic activity. Thus, the energy used by a neuron communicating with another neuron depends on the rate at which it fires spikes and the excitatory post synaptic potentials associated with each synapse. Thus, the number of other neurons with which a neuron communicates significantly affects the energy requirements. The relative proportion varies across species [45], but the focus in this article is on the spike activity. We show that spiking neural networks, even using the most basic approximation to the established Hodgkin-Huxley spike-dynamic equations [22], can exhibit significant energy savings in these inhibition models.

We consider two cases. The first implements our concept model of the previous paragraph. The second uses a Bayesian or attention approach to reduce energy costs even further. The essential feature of both models is the inhibition of inputs as soon as a concept has been activated.

To begin with, Section II surveys some of the related background work on the suppression of detail and energy constraints in the brain. Section III describes the simulation model. The methodology of the paper and the parameters for the simulation are given in Section IV. Section V provides theoretical analysis, Section VI gives results from the simulation and Section VII provides a detailed discussion and contextualisation. Section VIII concludes.

II. Related Work

Early indicators that some of this low-level detail might be accessible came from studies on victims of stroke and brain injury, where, for example, a person might discover the ability to draw realistically. Snyder and Mitchell [52] predicted that such access might be obtained using brain stimulation techniques, in which the conceptual part of the brain was blocked, because concepts inhibit lower level detail [51].

It transpired that this was indeed the case. The direct brain stimulation techniques, *Transcranial Magnetic Stimulation* (TMS) and the more recent technique, *Transcranial Direct-Current Stimulation* (TDCS) can be used to "switch off" part of the brain. By targeting the anterior temporal lobe in the left hemisphere—a brain area highly involved in concept formation and storage—it is possible to block access to concepts and thus release access to lower-level detail. In the first such study, now nearly a decade ago, drawing and proof-reading [53] were found to be enhanced by TMS. So, for example, it is hard for many people to see the word "the" when it is repeated on a following line. The ability to spot the error is enhanced when the meaning of the sentence is blocked by brain stimulation. Likewise numerosity [50] (rapidly estimating the number of objects in the field of view, and inspired by an incident in the film Rain Man), also goes up with TMS to the left anterior temporal lobe, i.e., blocking left temporal lobe activity. Over the subsequent decade, a diverse range of higher-level cognitive phenomena have been shown to be enhanced through dis-inhibition with brain stimulation. False memory, where like objects may get mixed up in memory tests (e.g., chair instead of stool), can be reduced [4]. Even the ability to solve visual puzzles can be enhanced [7].

There are numerous arguments for why this might be the case, such as the possibility of computational overload, discussed further in Section VII. In this era of information overload, such an explanation is at first sight appealing, but is hard to quantify with our existing knowledge of the brain.

Closely linked to computational overload is the energy cost of neural computation. The human brain uses about 20% of the body's energy [41] and various evolutionary changes, such as the appearance of meat in the diet, may have allowed the brain's energy consumption to grow. In fact estimates of brain energy usage per gram is about the same as human leg muscle during a marathon [2]. Navarette et al. [36] show that in over 100 species of mammal, adipose deposits correlate negatively with encephalisation. Thus, fat storage and increased brain size are oppositional strategies for avoiding starvation. But the cost of neural computation appears as a constraint across the animal kingdom. Plaçais and Preat show that in flies the brain disables costly memory mechanisms in the face of starvation [39].

Laughlin and Sejnowski [31] show that the brain's overarching network structure is consistent with preserving energy. The energy required for the transmission of nerve impulses, or spikes, and synaptic transmission are very tightly optimized, approaching the thermodynamic limits within cellular constraints [30]. The energy cost of transmission of a single bit of information turns out to cost around 10^4 ATP molecules. A single protein molecule, switching conformational states, was estimated to be able to switch at 1 ATP/bit, but the incorporation of this switch into the rest of the cellular circuitry is very costly.

The fundamental unit of energy across most animal systems is the energy released by conversion of the adenosine triphosphate (ATP) to adenosine diphosphate (ADP). This is about 10kT at human body temperature, where k is Boltzmann's constant and T is absolute temperature. The theoretical limit for transmission of a single bit would be kT, so the cellular cost is about 5 orders of magnitude above the theoretical limit.

Neuronal spikes account for a significant fraction of the neuronal energy usage [2].

The idea that the number of spikes might be kept to a minimum to save energy began with the idea of *sparse coding* in sensory systems [49][38]. More recently, cells have been observed, which fire strongly when the subject is exposed to stimuli corresponding to a particular person, say Bill Clinton, and to very little else [11][40]. They respond to the *concept,* and can be activated by pictures, voice or unique events. Obviously, for most people such a cell would fire very infrequently. The alternative distributed representation might have many cells coding for all US presidents. All of these cells would be active for any president, thus making their average activity much higher.

However, sparse coding is not the only way to reduce energy consumption by neurons using action potentials (APs). Changing the kinetics of the ion channels involved in generating the spike can reduce the energy requirements of the APs. Sengupta et al. [45] show that considerable differences in the relative cost of spike transmission versus the energy of synaptic transmission may be found, depending upon the exact ion channel kinetics, for example between giant squid neurons and those in mouse cortex.

This strong need to conserve energy suggests a possible explanation for why the raw sensory input is not accessible other than through external means such as TMS. *It is turned off to save energy.* Snyder et al. [51] and Bossomaier and Snyder [5] propose a *concept model* for how inhibition mechanisms might generate the observed effects of TMS. The effect is to turn off the inhibitory mechanisms, dis-inhibiting their targets.

Inhibition is of course widespread in the brain, and the prefrontal cortex—the area with most development over other primates—abounds in inhibitory effects. But, evidence is now emerging that even sensory perception in early areas such as primary visual area V1 depends upon top-down modulation, of which a large part is inhibitory [12][44].

From an evolutionary perspective, the human brain is more economical on energy use than other mammals. Lennie [32] calculates the rate of glucose (the brain's only source of metabolic energy) is three times lower than in rat and one and a half times lower in monkey. Thus, since the human brain is much bigger, this implies that fewer cells are active.

Feedback mechanisms are a common way of modulating input from lower processing areas of the cortex to higher processing areas. Visual processing streams provide a good example where higher order visual areas display an inhibitory top-down activity to lower visual processing areas such as V1 [12][44]. However, these models only consider connectivity patterns in the cortex related to visual processing. Jelinek and Elston [25] have shown that on a cellular level, processing complexity increases from V1 to prefrontal cortex, with layer-III pyramidal cell dendritic branching patterns becoming more complex and larger, thus requiring more energy. Higher visual processing areas deal more with conceptual phenomena by integrating simple information bits from lower processing areas.

Higher processing areas are not always necessary for rapid concept formation suggesting that single spike or a limited number of spikes representing different input to sensory areas can be already sorted prior to higher level processing and therefore enhancing processing speed. This has been shown in the auditory cortex. Higher cognitive processes, such as extracting similar patterns in varying acoustic input, or anticipation of acoustic input, already occur at the level of the sensory system rather than requiring higher cortical input. Thus, less information is passed on to higher levels such as the prefrontal cortex even if attention is not directed to the current sensory input [35].

In addition, reciprocal feedback has been shown between prefrontal areas of the cortex and hippocampus. Two pathways exist, which carry highly specific information from the prefrontal cortex to the hippocampus and the reciprocal connection allow rapid retrieval of information from the hippocampus by the prefrontal cortex [15].

Such top-down effects reduce activity at lower levels. Zhang et al. [56] show that in inferotemporal cortex, activity corresponding to a particular object is vastly different depending upon whether attention is focussed on that object.

III. The Simulation Model

The simplest approximation to the Hodgkin-Huxley equations is the Leaky Integrate and Fire (LIF) model. Izhikevich [23] points out that this neuron is capable of only a few of the 20, or so, behaviors, of which the full Hodgkin-Huxley model is capable. However, it is used here because *if a very simple model can generate the behavior we observe, then so can any of the more complex models.* This assures that the model is reasonably robust to parameter variations. Since more powerful neural models, such as the Izhikevich [23] model, can imitate the behaviour of simpler models (such as integrate and fire) then these more powerful models will have the same behaviour.

LIF has been used for very large scale models of the brain, such as that of Eliasmith [9]. Some limitations, which appeared in the simulations herein are discussed below.

Equation (1) shows the model for one neuron, where R is resistance, I the input current, u the membrane voltage and τ the time constant:

$$\frac{du}{dt} = -\frac{u}{\tau} + \frac{IR}{\tau} \tag{1}$$

Synaptic activation is represented by an alpha function with another time constant τ_s:

$$\varepsilon(t) = \frac{1}{\tau_s} e^{1-t/\tau_s} \tag{2}$$

The two simulation models use the same type of neuron, although the time constants are not the same.

A. Concept and Inhibition

Local inhibitory circuits have been studied in the visual system and hippocampus amongst others, which through inhibitory GABAnergic neurons and serotonergic neurons allow rapid information processing, in essence priming cells that leads to changes in membrane potential and more rapid response rates. Co-transmission and utilization of both G-protein coupled as well as ion channel gating enhances response rates. These structural specialisations presumably overcome synaptic delays in information processing due to neurotransmitter release, passage and binding between connected neurons [24], [17], [55].

In Model 1 we use a local inhibitory circuit, shown in Figure 1.

Since an eye fixation takes around 200 msec [29], we assume this represents the minimum time, for which a concept would remain active. The inhibitory circuit requires around 20 msec. It does not matter if input spikes come in as a single volley or as some Poisson process; if the maximum spike rate is around 100 spikes per second, the concept cell can see about 2 spikes in 20 msec, and should it see a spike from every cell, then it takes 40 msec to turn the input cells off. This would represent an spike saving of around a factor of 5.

B. Model 2: Prior Knowledge and Intention

There is abundant evidence of the use of Bayesian information processing throughout sensory and cognitive processing [26], [33]. For the purposes of this paper, the implication is that only a small subset of feature detectors need to fire to recognise something, given the assumption that something is going to appear.

Sometimes a single cue, such as hair color, might be enough to distinguish two people. So, if we know that the person coming up the driveway is one of two people who look very similar, then hair colour might be enough to distinguish them. In this case, it is not necessary to wait for all feature cells to fire. Just a few cells may suffice, in which case inhibition can start sooner. This is the essence of Model 2, illustrated in Figure 2. The prior neuron represents the assumption of what will appear: as soon as it has its minimal set of features it activates the output neuron, so, in turn, reducing the input activity early.

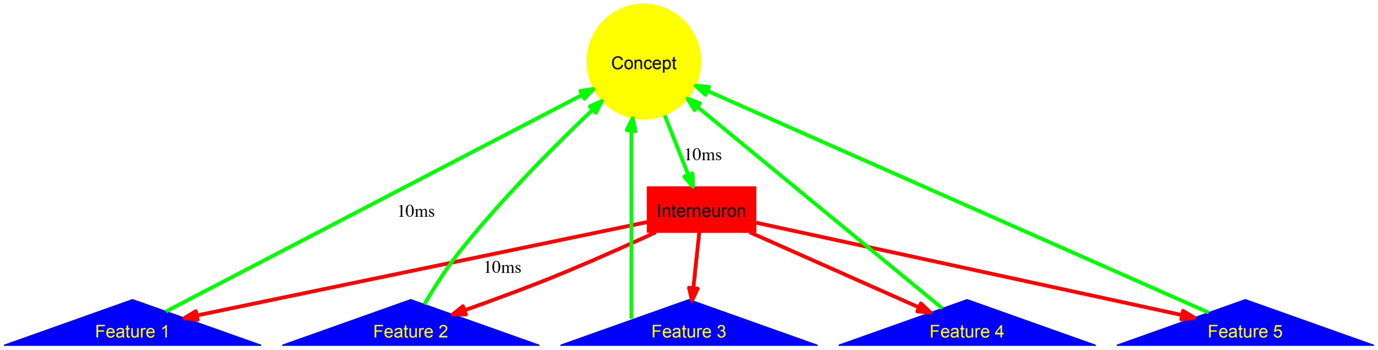


Fig. 1. The basic model. Sensory signals are the features in blue, of which there may be many more than 5. Connections from features to concept and concept to inhibitory interneuron are excitatory. Connections from the interneuron to the features are inhibitory. Parameters are in Table I

TABLE I
PARAMETERS FOR MODEL 1

neuron	τ	τ_s	R
Feature	1	2	20
Concept	50	50	20
Inhibitory	1	4	20

TABLE II
PARAMETERS FOR MODEL 2

neuron	τ	τ_s	R
Feature	1	2	20
Concept	50	50	10
Inhibitory	1	2	20
Prior	50	50	20

Now, assume that we have attentional control or a mindset that one is going to see K5 or K7 represented by the cells labelled *prior* in Figure 2. The facilitating cell is activated from higher up, but is agnostic as to whether K5 or K7 appears. It fires slowly with a long recovery time and brings a small subset of features closer to threshold. This costs a small number of spikes and synaptic events, since on average only one facilitating cell will fire. Now, only this small number of features needs to be activated for the concept to trigger. But, since these features lead over the remainder, only they will be allowed to fire.

IV. Methodology

We begin by carrying out a theoretical analysis of the possible energy savings in Section V. This makes use of a previously described framework [2] for estimating energy costs and compares the two models described above.

We then carry out computer simulations of both models. Simulations were carried out in Matlab using the Biological Neural Network toolbox [43]. The toolbox uses Matlab's integration routines for solving differential equations, for a variety of single neuron models. Since this is a proof-of-concept simulation, the choice of neuron model is not critical. Thus, the results presented here use the simple LIF model, described above.

Table I gives the parameters for the Model 1 simulation. Model 2 adds an additional neuron, the prior (in a Bayesian sense). Table II shows the parameters used for the simulation.

V. Theoretical Analysis

The most primitive biological model is Attwell and Laughlin's model A [2]. This is a concept-neuron cell model where x neurons sit latent and 1 neuron fires in response to the stimuli, for which it codes. This has an energy expense, E_A given by:

$$E_A = (x+1)r + a \tag{3}$$

where r is the resting energy usage, a the additional excitatory energy usage. For N concepts there are N neurons required.

Attwell and Laughlin's more complex model (B) is distributed representation, in which a subset of neurons represent each possible stimulus condition, or concept. Thus, we now have to look for the possible combinations, N_D, we can form If there are x inactive neurons while sets of y neurons encode a minimum number of conditions. i.e., x, y are integers such that the number of conditions, N_D is given by:

$$N_D = (x+y)!/(x!y!) \tag{4}$$

Its energy cost function, E_B, is given by:

$$(x+y)r + ya \tag{5}$$

Our Model 1 (denoted M1) has a concept neuron, an interneuron and a set n_i of feature neurons, the size of this set is $x = |n_i|$. For any concept, some number of feature neurons will fire to represent the concept, say eyes, nose and mouth to indicate a face. In the following analysis we make the simplifying assumption that the number of features is the same across all concepts. The count of the feature neurons x determines the maximum number of conditions that need to be encoded. A single concept neuron requires $m_i \subset n_i$ neurons to fire (i.e., $|m_i| = y, y < x$). Attwell and Laughlin [2] assert that there is a need to encode each concept as sparsely as possible, but now there is an overhead of $+2$ neurons (the concept and

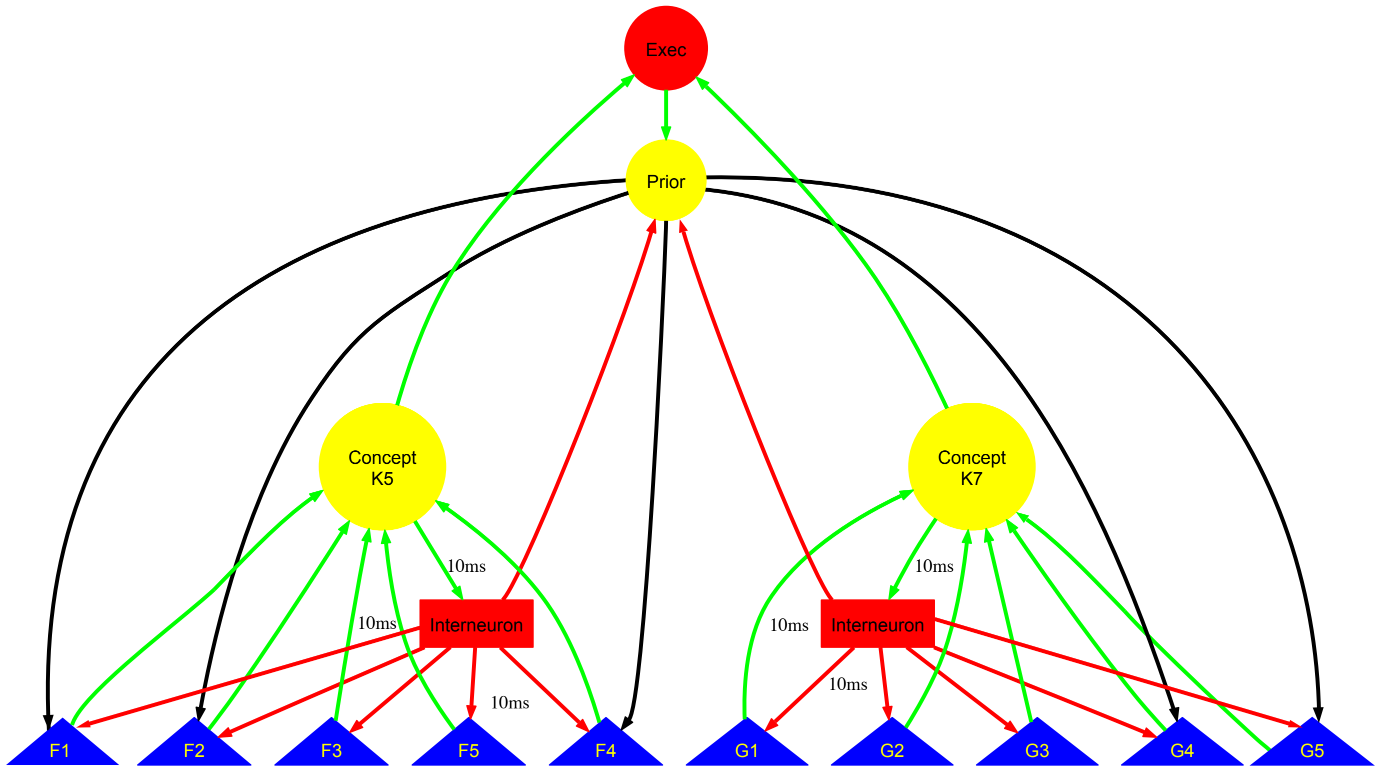


Fig. 2. The model with attention. Features, concepts and inhibitory interneuron are the same as in Model 1. Here we have two concepts and a single attention/prior neuron. This has excitatory connections to a small number of feature detectors.

the interneuron), but these two enforce the minimum sparse coding required.

$$x!/((x-y)!y!) \tag{6}$$

where y is the size of the minimal set of feature neurons necessary to encode a given concept. Its energy expense (Equation (5)) is now:

$$(x+y+2)r+(y+2)a \tag{7}$$

We assume that the energy costs of inhibition are the same as excitation, although they might be slightly less [2]. If $y >> 2$, i.e., there are many features needed to encode a concept sparsely, then the energy cost approximates Laughlin's Model B.

The Prior Knowledge Model, Model 2 (M2) model has a further overhead of a *prior neuron*. This adds additional neurons to the energistic overhead of M2 over and above Attwell and Laughlin's Model B. But the prior neuron also reduces the number of feature neurons that need to fire in order to activate the concept neuron. If the prior neuron is pre-exciting z feature neurons, and one feature is sufficient to uniquely distinguish between the other competing possibilities (such as hair colour or glasses etc.), then we can have optimal encoding using a prior neuron in the sense that two neurons, one encoding black hair and one encoding blonde hair, can encode the difference in the two concepts, so we (nearly) recover Attwell and Laughlin's optimal solution for the number of different, possible concepts, N_c

$$N_c = x!/((x-2)!2!) \tag{8}$$

(not conditions) for a cost, E_p, of

$$E_p = (x+1+4)r+(4+1)a \tag{9}$$

Note that the interneuron will still fire in this model, this increases the fidelity of the signal propagating to the concept neuron by suppressing possible confounding signals (just as it usually does). Although in this paper a single prior neuron is used, the pre-narrowing of a set of conditions, in a Bayesian framework is a well substantiated model of perception [26], [33].

Returning to the example above, where the selected feature is to distinguish between either black or blonde hair, this feature is encoded in two separate feature neurons that connect to two different concept neurons. When the prior pre-excites these two feature neurons they need to provide a voltage potential to the axon hillock sufficient to fire the concept neuron by themselves. If all feature neurons are connected to a concept neuron with the same synaptic strength, then doubling the firing rate of a single feature neuron doubles the energy that neuron expends but only provides the same amount of voltage potential (less some leakage) to the axon hillock as two other feature neurons. If, however, it is a strongly connected neuron, then doubling its firing rate may equate to adding the equivalent voltage potential of four lesser connected feature neurons.

We can make this a stronger statement. The prior neuron pre-excites a feature neuron by reducing the potential threshold the feature neuron needs to achieve in order to fire. For example, a useful ballpark figure of 100mV is needed to fire an excitatory neuron, a prior neuron can provide the first 80mV in the form an excitatory voltage that essentially lowers the threshold in the feature neuron to only 20mV. This means the feature neuron fires $m = 5$ times more readily than before (assuming a linear integrate and fire profile with limited neural voltage leakage), i.e.,

$$m = \frac{100mV}{(100mV - 80m)}V \tag{10}$$

If the prior neuron continues to provide this excitatory signal then potentially the feature neuron could fire 5 times more frequently (original frequency times 5), again, assuming linearity and limited leakage.

Energy is saved only if the synaptic weights connecting the feature neuron to the concept neuron are inhomogeneous. To see this assume that the pre-excited feature neuron has a synaptic weight $w_1 = 1$ and there are n other neurons also connected to the concept neuron with weights

$$w_i = \frac{1}{(n+k)} \tag{11}$$

for some k a free parameter, as discussed below. Assume that all $n+1$ neurons have approximately the same noisy, leaky integrate and fire profile (weakly leaking and linearly integrating over time). Then each neural pulse received by the concept neuron from the strongly connected and pre-excited feature neuron is stronger than the weakly connected neurons by a factor, q, given by:

$$q = \nu_{nat}\frac{100mV}{(100mV - v_{pre})}\frac{w1}{(n+k)} \tag{12}$$

with ν_{nat} a natural frequency. and v_{pre} the pre-excited potential. This simply says that a neural pulse received by the concept neuron from the pre-excited feature neuron is both stronger and occurs more frequently than the signal from the weakly connected neurons. As $k \to 0$ the expected firing of the feature neurons that are not pre-excited has an expected aggregate contribution to the firing of the concept neuron that is equivalent to the signal provided by the pre-excited neuron. At the point $k = 0$ it is just as likely that all feature neurons have fired as it is the pre-excited feature neuron fires, indicating the onset of an energy usage transition point where many neurons start firing at $k < 0$ because the aggregate signal from the rest of the neural population begins to fire before the pre-excited neuron. The neurons that are not pre-excited are unlikely to fire when $k > 0$ and there is an energy saving of na non-firing neurons where a is the additional excitatory (firing) energy over and above the resting energy as introduced earlier. The cost of this is two neurons (the prior neuron and the pre-excited neuron) that fire with an increased rate of:

$$\nu_{nat}\frac{100mV}{100mV - v_{pre}} = \Delta f \tag{13}$$

for a total cost of

$$E_{total} = 2\Delta f \tag{14}$$

Note that the term $\frac{100mV}{100mV - v_{pre}}$ is the ratio of the un-excited neural firing frequency to the pre-excited neural firing frequency.

Bringing all of this together we can discuss the energy savings available in an example. Assume that 1,000 feature neurons encode two different concepts, each concept being uniquely composed of 500 features each, each concept has inhibitory interneurons. We consider two examples, one, in which there is a prior, and one, in which there is not:

- **No prior neuron:**
  500 feature neurons contribute a noisy, linear integrate and fire signal to each of the concept neurons. The concept neuron that fires first has its winner takes all signal passed directly to the executive. At the same time the interneurons are activated and inhibit 450 feature neurons from firing. In this case M1 saves $a(450 - 2)$ ATP units per concept over the simpler model of single neurons encoding whole concepts in an excitatory fashion. A total of $2 \times 52a$ ATP are used by excited neurons, the factor of two accounts for the energy usage for both concepts.
- **Prior neuron:** The prior neuron pre-excites two neurons, one for concept neuron A and one for concept neuron B, pre-excitement occurs before any other external information excites the feature neurons. When information does impinge upon the feature neurons, pre-excitement increases the frequency of firing of both feature neurons by a factor of 5, i.e., the firing threshold was 100mV, it is now 20mV for both neurons. This costs $2\Delta fa = 10a$ for both concept neurons, i.e., $20a$ total because pre-excitement instigated firing happens regardless of whether or not the concept neuron fires. The neural savings achieved through this cost is $2a$ ATP , the net savings in energy consumption is $(88+1)a$ ATP (one of the concept neurons does not fire), nearly an order of magnitude better than without a prior neuron.

The above analysis is minimalist, but still requires a few assumptions:

- It requires the connectivity between the non-pre-excited feature neurons to be lower than the inverse of the number neurons. Without this assumption the prior neuron is unable to boost the signal from the pre-excited neuron sufficiently to activate before the rest of the feature neuron population can activate.
- The value of a is arbitrary, but Attwell and Laughli [2] give estimates for rat grey matter. All calculations are relative, relative frequencies and relative energy consumption, as this is a purely comparative analysis. The

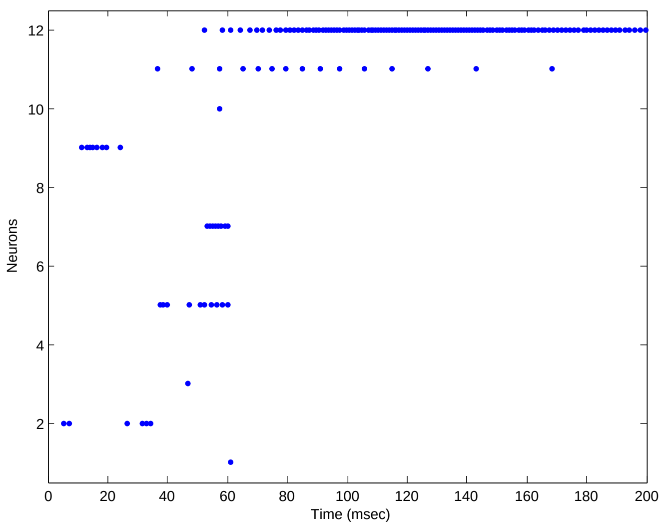


Fig. 3. Activity of the network of model 1. Cells are laid out along the y-axis. The top cell is the inhibitory interneuron, the next cell down is the concept and the remainder are the features. Each dot represents a spike event. The inhibition in neuron 12 sets in after the concept neuron has started to fire.

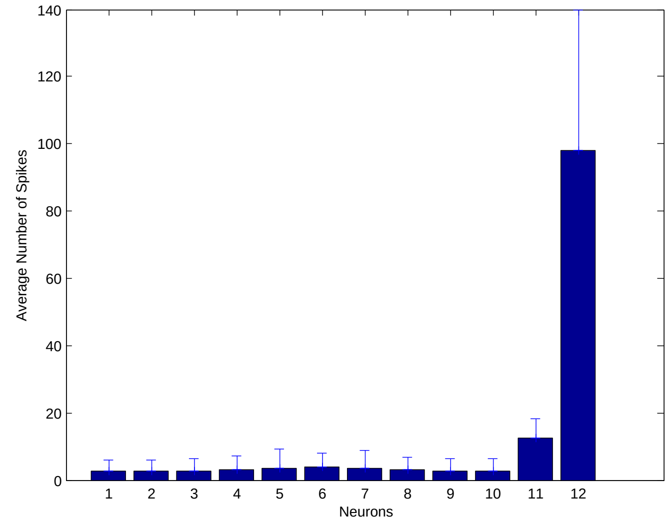


Fig. 4. The average number of spikes for each neuron. Neurons 1–10 are the input features, neuron 11 the concept and neuron 12 the inhibitory neuron.

actual ratio $\frac{r}{a}$ depends on the temporal resolution, which determines the firing rate, R. For rat this varies by an order of magnitude from $a = 6.4R$ at a low firing rate of 4Hz to $a = 64R$ at a high firing rate of 40Hz, with an approximately linear increase of energy usage with firing rate.

- There has been no explicit discussion of the synaptic costs, with a focus on action potentials. The costs vary depending upon species, but can be as much as a third of the energy budget [2].

In both M1 and M2, the concept neurons are single cells, but this is not contrary to the efficient distribued model. The inhibitory interneurons may be activated by a collection of cells (a distributed concept). But for simplicity we have confined the model to just one or two concept cells.

VI. Simulation Results

Figure 3 shows the spiking patterns for Model 1. The features are suppressed for the duration of activation of the concept, representing at least a substantial decrease in energy usage. Whereas the activity of the concept and inhibitory neurons are maintained throughout the 200msec simulation, activity of the feature neurons rapidly dies away. Without the inhibition, their firing would also be maintained. Figure 4 shows the average number of spikes in each neuron over 100 runs.

Turning to Model 2, Figure 5 shows the activity with input (from the executive) to the prior neuron only. There is no activity from the feature neurons showing that they do not get over threshold from the prior alone.

Figure 6 shows the effect of adding a small input to the feature detectors. The neurons sensitised by the prior (2,8,10, here, but selected at random in each run) now fire and activate the concept neuron (number 11).

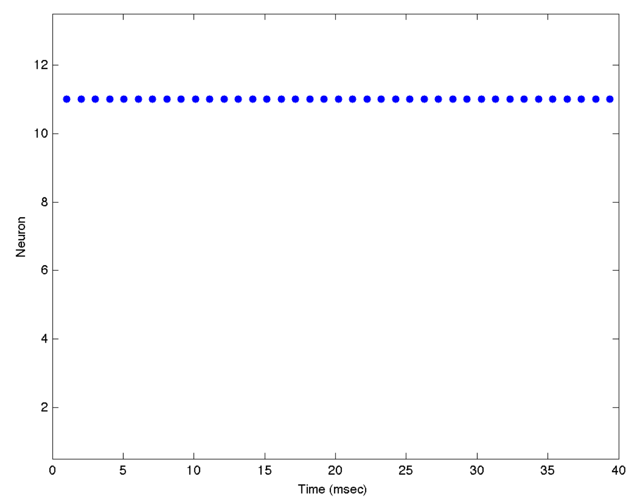


Fig. 5. Input to prior neuron only. The feature neurons do not get over threshold

Figure 7 shows increased input to the feature neurons, which now all get over threshold and fire.

Finally, Figure 8 introduces the inhibitory neuron. The feature neurons and the prior are suppressed. The concept neuron stops firing as the input dries up.

In this article, only one concept neuron appears, but a single prior could in principle pre-activate any number of feature neurons, subserving more than one concept, but biasing the outcome to some subset of possible concepts, which might occur in a given context.

VII. Discussion

The simulation model presented here demonstrates the feasibility of saving energy through inhibition of lower levels. Since this is a new conjecture to the best of the authors' knowledge,

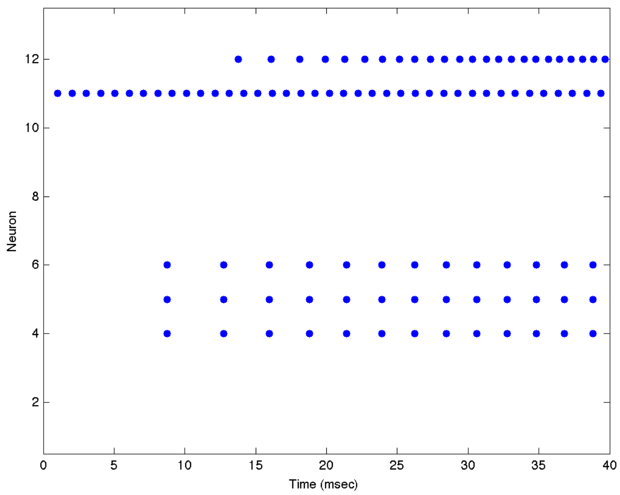


Fig. 6. Simulation with low input to the feature neurons, enough to drive those pre-sensitised by the prior over threshold

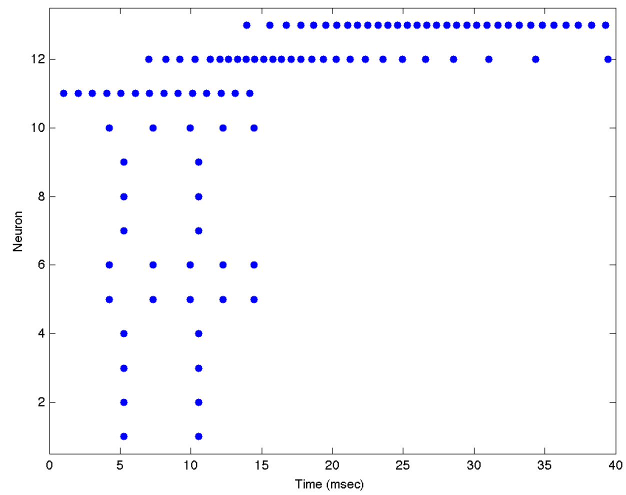


Fig. 8. Full Model 2 with prior neuron and inhibition

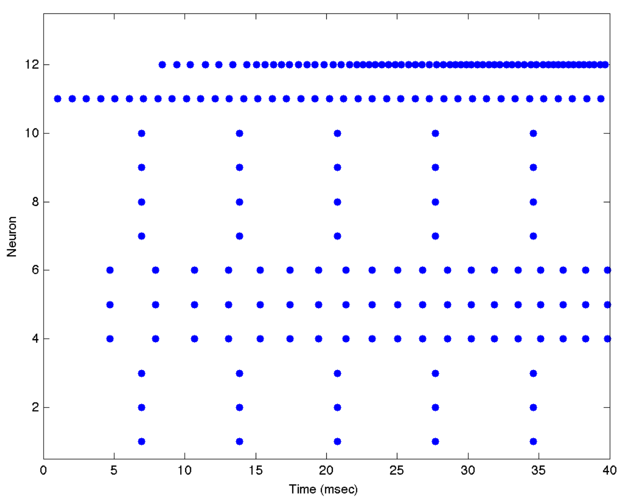


Fig. 7. High input to feature neurons resulting in all of them getting over threshold. Higher firing rate is seen for the prior-sensitised neurons.

there are no prior results for comparison. The model uses the simplest spiking neuron, on the basis that any more complex model would be able to reproduce the same effects if they exist for the LIF neuron.

However, one limitation of this neuron did become apparent. Ideally the concept neuron should be able to maintain its firing rate until attention switches to something else. However, to maintain firing rate in LIF would require increasing τ_s. This has the consequence of either causing the concept firing to build very slowly, or to allow the firing rate to grow unnecessarily high. The simple LIF does not saturate easily. Further work would look at the more sophisticated Izhikevich models [23].

The conjecture that it is possible to reduce the spikes generated by a feature might seem surprising. There is, however, substantial work demonstrating that a single spike per neuron may be enough for pattern recognition. Thorpe et al. [54] discovered that people can make *very* rapid decisions on whether pictures contain animals – so rapid that they are likely to be able to use only a single spike along the path from retina to associative cortex. Subsequent computational models demonstrate the feasibility of the single spike model.

The information overload argument suffers from a lack of understanding of what the brain can actually do on a large scale. We know something about the capacity of simple neural networks, such as the number of patterns storable in a Hopfield net or the Vapnik-Chervonenkis Dimension of feedfoward networks. But on the scale of the cortex we have only the most rudimentary of measures.

There has been some interesting progress made in understanding the macroscopic (i.e., psychological) aspects of such limits in real cognitive processes. As early as 1956 Miller [34] proposed on information theoretical grounds that our ability to store and retrieve information is limited. Recent work has extended this idea for complex games [21] showing that *chunks* have a finite information capacity that can be inferred from game data. Chunks can be thought of approximately as the feature neurons of the cognitive models discussed in this work. These chunks have been extensively studied by Simon, Gobet and colleagues [48], [47], [42], [6] in order to explain the rapid, efficient and quite remarkable talents of world leading experts in diverse specialist domains such as chess, nursing and physics.

Beyond these chunks a further mechanism, called *templates* has been suggested as a way to aggregate and contextualise chunks [13], [16], [19], [20]. Using these notions of chunks and templates there is some empirical evidence to suggest that experts may pass through transition points in their development [18] as first suggested on theoretical grounds by Ericsson [10]. These cognitive results provide high level principles with, which to inform the neuro-cognitive models such as that presented in this work.

It is of no small significance that the prior neuron as described in this work might be identifiable with a *concept neuron* (or a small set of neurons encoding the same or a very similar concept). Recent neurological studies on monkeys and humans [28] have shown that categories of objects, candidates for the instantiation of templates, have a remarkable similarity between species. A theoretical model [46] of feedforward processes for generating and implementing such categories or concepts has been put forward for more general processes such as searching a scene or recognising objects. Such neurological and psychological studies continue to provide the substantive empirical support for the theoretical frameworks such as the one proposed in this work.

Darwin [8] famously remarked: *to suppose that the eye [...] could have been formed by natural selection, seems, I freely confess, absurd in the highest degree.* A century later, Nillson and Pelger [37] showed that evolving an eye was actually relatively easy. By the same token, without a very good model of the computational limits of the brain, the information-overload argument is hard to substantiate.

On the other hand, people are good at blocking out stimuli. The noise of a busy road, the drone of the engines in an aircraft cabin, the buzz of other speakers in a cocktail party – all demonstrate our remarkable capacity to shut out interference when we so desire. But this blocking is reversible and we *can* turn our attention to the distractions themselves. Koechlin [27] shows that the pre-frontal cortex can select one context and block others in choosing an action.

A model that allow specific context retrieval that combines alternate views of top-down guidance by prefrontal cortex or selection of goal-relevant information has been proposed by Badre et al. [3] It proposes that different areas of the prefrontal cortex are specialised for different functions that combine to enhance cognitive processing speed.

The blocking of sensory detail seems to be hardwired and is *not* switchable. To turn off this inhibition would require additional circuits to turn off conceptual information. In general, such circuits do not seem to have evolved, and external techniques such as TMS are required for their inhibition. This would make sense: strategies to save energy would be likely to have evolved much earlier than the expansion of the cortex and its sophisticated filters and control mechanisms.

VIII. CONCLUSIONS

One of the most remarkable findings of the last two decades has been the discovery of the way higher level concepts inhibit low level detail in most individuals, although not in high-functioning autistic savants. The building of conceptual structures on top of the the raw sensory detail is essential for advanced cognition, and is illustrated by the many difficulties experienced in Asbergers and autism, where this conceptual building seems to be impaired [53].

This paper shows that there is an energetic cost to maintaining access to this this raw detail. Thus, the reason we cannot have both a conceptual system and a savant-like raw detail system, might arise from the need to conserve energy.

REFERENCES

[1] T. Bossomaier, L. Barnett, V. Thiruvarudchelvan, and H. Jelinek, “Energy saving accounts for the suppression of sensory detail,” in *Proc. Fourth International Conference on Advanced Cognitive Technologies and Applications, COGNITIVE12, Nice*, 2012, p. 14–18.

[2] D. Attwell and S. B. Laughlin, “An energy budget for signaling in the grey matter of the brain.” *J. Cereb. Blood Flow Metab.*, vol. 21, pp. 1133–1145, 2001.

[3] D. Badre, R. Poldrack, E. Paré-Blagoev, R. Insler, and D. Wagner, “Dissociable controlled retrieval and generalized selection mechanisms in ventrolateral prefrontal cortex,” *Neuron*, vol. 47, pp. 907–918, 2005.

[4] P. S. Boggio, F. Fregni, C. Valasek, S. Ellwood, R. Chi, J. Gallate, A. Pascual-Leone, and A. W. Snyder, “Temporal lobe cortical electrical stimulation during during the encoding and retrieval phase reduces false memories,” *PLoS One*, vol. 4, no. 3, p. e4959, 2009.

[5] T. Bossomaier and A. W. Snyder, “Absolute pitch accessible to everyone by turning off part of the brain?” *Organised Sound*, vol. 9, pp. 181–189, 2004.

[6] P. Chassy and F. Gobet, “Visual search in ecological and non-ecological displays: Evidence for a non-monotonic effect of complexity on performance,” *PloS one*, vol. 8, no. 1, p. e53420, 2013.

[7] R. P. Chi and A. W. Snyder, “Facilitate insight by non-invasive brain stimulation,” *PLoS One*, vol. 6, no. 2, p. e16655, 02 2011.

[8] C. Darwin, *On the Origin of the Species*. John Murray, 1859.

[9] C. Eliasmith, T. Stewart, X. Choo, T. Bekolay, T. DeWolf, C. Tang, and D. Rasmussen, “A large-scale model of the functioning brain,” *Science*, vol. 338, pp. 1202–1205, 2012.

[10] K. A. Ericsson and N. Charness, “Expert performance: Its structure and acquisition.” *American Psychologist*, vol. 49, no. 8, p. 725, 1994.

[11] K. Gaschler, “One person, one neuron?” *Scientific American*, vol. 17, pp. 77–82, 2006.

[12] C. Gilbert, M. Ito, M. Kapadia, and G. Westheimer, “Interactions between attention, context and learning in primary visual cortex,” *Vision Research*, vol. 40, pp. 1217–1226, 2000.

[13] F. Gobet, P. C. Lane, S. Croker, P. C. Cheng, G. Jones, I. Oliver, and J. M. Pine, “Chunking mechanisms in human learning,” *Trends in Cognitive Sciences*, vol. 5, no. 6, pp. 236–243, 2001.

[14] F. Gobet, A. Snyder, T. Bossomaier, and M. Harr, “Designing a better brain: Insights from experts and savants,” *Frontiers in Psychology*, vol. 5, no. 470, 2014. [Online]. Available: http://www.frontiersin.org/cognition/10.3389/fpsyg.2014.00470/full

[15] P. S. Goldman-Rakic, L. D. Selemon, and M. L. Schwartz, “Dual pathways connecting the dorsolateral prefrontal cortex with the hippocampal formation and parahippocampal cortex in the rhesus monkey,” *Neurosci.*, vol. 12, no. 3, pp. 719–743, 1984.

[16] A. Guida, F. Gobet, H. Tardieu, and S. Nicolas, “How chunks, long-term working memory and templates offer a cognitive explanation for neuroimaging data on expertise acquisition: A two-stage framework,” *Brain and Cognition*, vol. 79, no. 3, pp. 221–244, 2012.

[17] Z.-S. Han, E. H. Buhl, Z. Lrinczi, and P. Somogyi, “A high degree of spatial selectivity in the axonal and dendritic domains of physiologically identified local-circuit neurons in the dentate gyms of the rat hippocampus.” *Eur. J. Neurosci*, vol. 5, pp. 395–410, 1993.

[18] M. Harré, T. Bossomaier, and A. Snyder, “The development of human expertise in a complex environment,” *Minds and Machines*, vol. 21, no. 3, pp. 449–464, 2011.

[19] ——, “The perceptual cues that reshape expert reasoning,” *Nature Scientific Reports*, vol. 2, 2012.

[20] M. Harré and A. Snyder, “Intuitive expertise and perceptual templates,” *Minds and Machines*, pp. 1–16, 2012.

[21] M. Harré, T. Bossomaier, A. Gillett, and A. Snyder, “The aggregate complexity of decisions in the game of go,” *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 80, no. 4, pp. 555–563, 2011.

[22] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *Journal of Physiology*, vol. 117, pp. 500–544, 1952.

[23] E. Izhikevich, “Which model to use for cortical spiking neurons,” *IEEE Trans. Neural Networks*, vol. 15, pp. 1063–1070, 2004.
[24] R. L. Jakab and P. Goldman-Rakic, “Segregation of serotonin 5-HT2A and 5-HT3 receptors in inhibitory circuits of the primate cerebral cortex,” *J. Comparative Neurology*, vol. 417, no. 3, pp. 337–348, 2000.
[25] H. F. Jelinek and G. N. Elston, “Dendritic branching of pyramdial cells in the visual cortex of the nocturnal owl monkey: A fractal analysis,” *Fractals*, vol. 11, no. 4, pp. 391–396, 2003.
[26] D. Knill and W. Richards, *Perception as Bayesian Inference*. Cambridge University Press, 1996.
[27] E. Koechlin, C. Ody, and F. Kouneiher, “The architecture of cognitive control in the human prefrontal cortex,” *Science*, vol. 302, no. 5648, pp. 1181–1185, 2003.
[28] N. Kriegeskorte, M. Mur, D. A. Ruff, R. Kiani, J. Bodurka, H. Esteky, K. Tanaka, and P. A. Bandettini, “Matching categorical object representations in inferior temporal cortex of man and monkey,” *Neuron*, vol. 60, no. 6, p. 1126, 2008.
[29] M. Land and B. Tatler, *Looking and Acting: Vision and Eye Movements during Natural Behaviour*. Oxford University Press, 2009.
[30] S. B. Laughlin, R. d. Ruyter van Steveninck, and J. Anderson, “The metabolic cost of neural computation,” *Nature Neuroscience*, vol. 1, no. 1, pp. 36–41, 1998.
[31] S. B. Laughlin and T. J. Sejnowski, “Communication in neural networks,” *Science*, vol. 301, no. 5641, pp. 1870–1874, 2003.
[32] P. Lennie, “The cost of cortical computation,” *Current Biol.*, vol. 13, no. 493–497, 2003.
[33] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget, “Bayesian inference with probabilistic population codes,” *Nat. Neurosci*, vol. 9, no. 11, pp. 1432–1438, 2006.
[34] G. A. Miller, “The magical number seven, plus or minus two: some limits on our capacity for processing information.” *Psychological review*, vol. 63, no. 2, p. 81, 1956.
[35] R. Näätänen, M. Tervaniemi, E. Sussman, P. Paavilainen, and I. Winkler, “Primitive intelligence in the auditory cortex,” *Trends in Neurosci*, vol. 24, no. 5, pp. 283–288, 2001.
[36] A. Navarette, C. P. v. Schaik, and K. Isler, “Energetics and the evolution of human brain size,” *Nature*, vol. 480, pp. 91–94, 2011.
[37] D.-E. Nillson and C. Pelger, “A pessimistic estimate of the time required for an eye to evolve,” *Proc. Royal Soc. Lond. B*, vol. 256, pp. 53–58, 1994.
[38] B. Olshausen and D. Field, “Sparse coding with an overcomplete basis set: A strategy by V1?” *Vision Research*, vol. 37, no. 3, pp. 3311–3325, 1997.
[39] P.-Y. Pla¸cais and T. Preat, “To favor survival under food shortage, the brain disables costly memory,” *Science*, vol. 339, pp. 440–442, 2013.
[40] R. Q. Quiroga, L. Reddy, G. Kreiman, C. Koch, and L. Fried, “Invariant visual representation by single neurons in the human brain,” *Nature*, vol. 435, pp. 1102–1107, 2005.
[41] M. E. Raichle and D. A. Gusnard, “Appraising the brain’s energy budget,” *PNAS*, vol. 99, no. 16, pp. 10 237–10 239, 2002.
[42] H. B. Richman, F. Gobet, J. J. Staszewski, and H. A. Simon, “Perceptual and memory processes in the acquisition of expert performance: The KP AM Model,” in *The Road to Excellence*, K. Ericsson, Ed. Mahwah, NJ: Erlbaum, 1995.
[43] A. Saffari, “Biological Neural Network Toolbox for Matlab.” [Online]. Available: http://www.ymer.org/amir/software/biological-neural-networks-toolbox
[44] R. Schäfer, E. Vasilaki, and W. Senn, “Perceptual learning via modification of cortical top-down signals,” *PLoS Comput Biol*, vol. 3, no. 8, p. e165, 2007.
[45] B. Sengupta, M. Stemmler, S. B. Laughlin, and J. Niven, “Action potential energy efficiency varies among neuron types in vertebrates and invertebrates,” *PLoS Comput Biol*, vol. 6, no. 7, p. e1000840, 07 2010.
[46] T. Serre, A. Oliva, and T. Poggio, “A feedforward architecture accounts for rapid categorization,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 15, pp. 6424–6429, 2007.
[47] H. A. Simon, “What is an “explanation” of behavior?” *Psychological Science*, vol. 3, no. 3, pp. 150–161, 1992.
[48] H. A. Simon and W. G. Chase, “Skill in chess: Experiments with chess-playing tasks and computer simulation of skilled performance throw light on some human perceptual and memory processes,” *American scientist*, pp. 394–403, 1973.
[49] E. Simoncelli and B. Olshausen, “Natural images statistics and neural representation,” *Annual Rev. Neurosci*, vol. 24, pp. 1193–1216, 2001.
[50] A. W. Snyder, H. Bahramali, T. Hawker, and D. Mitchell, “Savant-like numerosity skills revealed in normal people by magnetic pulses,” *Perception*, vol. 35, no. 6, pp. 837–845, 2006.
[51] A. W. Snyder, T. Bossomaier, and D. Mitchell, “Concept formation: object attributes dynamically inhibited from conscious awareness,” *Journal of Integrative Neuroscience*, vol. 3, pp. 31–46, 2004.
[52] A. W. Snyder and D. J. Mitchell, “Is integer arithmetic fundamental to mental processing?: The mind’s secret arithmetic,” *Proc. Royal Soc. London B*, vol. 266, pp. 587–592, 1999.
[53] A. W. Snyder, E. Mulcahy, J. L. Taylor, D. J. Mitchell, P. Sachdev, and S. Gandevi, “Savant-like skills exposed in normal people by supressing the left fronto-temporal lonbe,” *Journal of Integrative Neuroscience*, vol. 2, no. 2, 2003.
[54] S. Thorpe, A. Delorme, and R. v. Rullen, “Spike-based strategies for rapid processing,” *Neural Networks*, vol. 14, pp. 715–725, 2001.
[55] R. Yuste, J. N. MacLean, J. Smith, , and A. Lansner, “The cortex as a central pattern generator,” *Nat. Rev. Neurosci*, vol. 6, no. 6, pp. 477–483, 2005.
[56] Y. Zhang, E. Meyers, N. Bichot, T. Serre, T. Poggio, and R. Desimone, “Object decoding with attention in inferior temporal cortex,” *PNAS*, vol. 108, pp. 8850–8855, 2011.

Single-Handed Eyes-Free Chord Typing: A Text-Entry Study

Adrian Tarniceriu, Pierre Dillenbourg, and Bixio Rimoldi

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
Email: adrian.tarniceriu@epfl.ch, pierre.dillenbourg@epfl.ch, bixio.rimoldi@epfl.ch

***Abstract*—For most users, interacting with mobile computing devices requires visual commitment to the input mechanism. As a result, devices such as smartphones and PDAs are not suitable in situations when visual attention is already focused on another task. Chording devices do not have this drawback but require some training. We evaluate the performance of a key-to-character mapping for a 5-key chording device designed to minimize the learning phase. The subjects in our study were able to memorize the mapping in the first 45 minutes of training. After approximately 350 minutes, the average entry speed was 20 words per minute. The influence of having visual, audio or no feedback was also evaluated. We found that the typing rates were the same under all three conditions, but the error rates were the smallest in the absence of feedback (2.32%) and the largest when the users could see what they typed (3.41%).**

Keywords–chording keyboard, text entry, key mapping, feedback.

I. INTRODUCTION

There are currently many methods for interacting with mobile computing devices; the most popular ones are graphical interfaces, voice command and text input. Graphical interfaces are probably the most user-friendly, but they require the user's visual attention, which makes them unsuitable in situations where vision is already committed, such as walking in a crowded place, pushing a shopping cart, riding a bike, and driving a car. Yet, it would be nice if we could send a quick note while walking to work, or if we could look up the characteristics and the location of an item while pushing a shopping cart. While exerting physical activities such as jogging or riding a bike, the smart phone could process biometric information and could respond to various queries. Even though more controversial, it would be nice if, while driving a car, we could issue commands like "find fastest way home" or "inform partner of arrival time". If we could issue such commands safely, there would be no problem for the on-board computer to estimate the arrival time and interact with the driver's smartphone to send a message to the partner. If we could issue such commands with voice, few people would be concerned about safety, but there are many problems associated with using voice to control a computer: the performance is not satisfactory, particularly in noisy places, there is an issue with privacy, and there are moments/places where other people would not appreciate hearing us speak aloud to our computer. Brain computer interfaces would solve all these problems, but the technology is not sufficiently developed yet. Although text is not the most user-friendly way to interact with a mobile device, it could be the most efficient in the aforementioned situations if we could type without looking at the keyboard and without a significant extra cognitive load.

This paper extends our previous work [1], [2] by presenting a study on a 5-key chording device that, we believe, has high potential for application in all of the above mentioned situations. This type of keyboard enables users to generate a character by simultaneously pressing a combination of keys, similarly to playing a note on a musical instrument. With five keys, there are 31 combinations in which at least one key is pressed, enough for the 26 letters of the English alphabet and five other characters. If the keys are in a position that is naturally under the fingertips, a person can type using the fingers of one hand, without committing the eyes to the keyboard. For instance, by placing the keys on the handlebar of a bike, we can control the bike with both hands and type at the same time, because typing only requires varying the pressure under the fingertips. Some visual (or auditive) feedback is still needed occasionally to verify the output and correct eventual mistakes, but this requires considerably less commitment than continuously looking at the input device. The required visual attention can be further reduced by displaying the output in the natural field of vision, for instance on a windshield or on goggles.

The main drawback of a chording device is that before being able to use it, one should learn the correspondence between key combinations and characters. We present a mapping designed to minimize the learning time by assigning intuitive combinations to each character, and a study that evaluates the proposed mapping. We will evaluate the achievable typing rates, the error rates, the characters that are more difficult to type, and the distribution of typing errors. Afterwards, we will analyze how different types of feedback (visual, auditive, and no feedback) affect the ability to type with a chording keyboard.

The paper is organized as follows. In Section II, we present a brief overview of related work. In Section III, we describe a key-to-character mapping for a 5-key keyboard designed to reduce the learning time. We denote this mapping in the following as 5keys. In Section IV, we present an experiment that evaluates the learnability of the mapping, and in Section V, we evaluate the achievable text-entry rates, typing accuracy, common error patterns, and three different feedback types. In Section VI, we conclude the paper and discuss future directions.

II. Related Work

In order to make a keyboard suitable for mobile technologies, we can make the keys very small and/or remove the one-to-one mapping between keys and characters [3]. The first method includes mini-QWERTY, on-screen keyboards, or RearType [4], and the second method includes multi-tap (with or without T9), LetterWise [5], TiltText [6], FrogPad [7], or chording keyboards. Another possibility is given by gestural text-entry techniques such as Graffiti [8], Edgewrite [9], and the minimal device independent text input method (MDITIM) [10]. Dynamic selection techniques such as Dasher [11] use probabilistic techniques to make the most likely characters or sets of characters easier to type, based on the already typed text.

Chording keyboards were first used in stenotype machines (starting from the 1830s) and telegraph communications. One such example is the Baudot code [12] (patented in 1874) that assigns five bits to a character and evolved into the International Telegraphy Alphabet No. 2, still used by some radio amateurs. Another application is represented by the Braille system, that enables blind people to read and write. Subsequent studies were performed by IBM, where researchers developed both single-handed and two-handed keyboards, with the number of keys ranging from 8 to 14 [13]. However, research stopped in 1978. Douglas Engelbart, the inventor of the computer mouse, also proposed a 5-key keyset, but this was not incorporated in any system [14]. Microwriter [15] was a chording portable word processor commercialized in the early 1980s, but again, it was not a commercial success.

As traditional desktop applications such as text editors, schedulers or e-mails have become available on mobile devices, there has been a significant increase in text entry research. This also lead to renewed interest in chording keyboards and to the appearance of several new devices: DataEgg, appeared in the early 1990s, is a 7-key handheld device with pager, phonebook, e-mail and calendar functions [16]. GKOS (2000) is a 6-key two-handed input device that can be used for text input or game control [17]. Chordite (2002), Twiddler (2004) and EkaPad (2009) are pocket sized, single-handed keyboards that can also have miniature joystick or mouse-like abilities [18], [19], [20]. The chording glove [21] is a chord keyboard where the buttons are mounted directly on the fingers. Typing studies involving chording keyboards include those performed by Lyons et al. for the Twiddler, [3], [19], [22], and by Rosberg and Slater for the chording glove [21]. Sandnes et al. propose a chording interface for controlling in-car devices such as music player, navigation system, lights or telephone [23], and an error correction mechanism for three and five-key chording keyboards [24], [25].

In mobile environments, users cannot usually look at the text-input device and/or at the display while typing; this condition is denoted as "blind" or "eyes-free" typing [26]. Therefore, it is important to analyze how visual feedback affects the text-entry process. Silfverberg examined the effect of both tactile and visual feedback when using mobile phone keypads [26] and found that reduced tactile feedback increases the typing error rate. In addition, low visual feedback also leads to more errors, decreasing accuracy. A similar study made by Clawson et al. [27], concerning typing with mini-QWERTY keyboards, demonstrates the importance of seeing the keys while typing. However, no significant differences in typing speeds and error rates were noticed when users could or could not see the typed text.

The above studies stress the importance of seeing the input device in the case of 4×3 multi-tap keypads and mini-QWERTY keyboards. But this should not be an issue for most chording keyboards, that are specifically designed to be operated without looking at the keys. Typing experiments with limited visual feedback for the Twiddler chording keyboard were performed by Lyons et al. [22], and show that, surprisingly, typing and error rates actually improve with reduced visual feedback. Mascetti et al. propose and evaluate a Braille typing system for smartphones [28]. As it is intended for visually impaired persons, there is no visual, but only audio feedback.

Other studies where participants do not look at the typing device or are involved in dynamic activities that require vision commitment include the already mentioned chording glove [21], a two-handed chorded software keyboard for PDAs [29], half-QWERTY touch typing [30], or the keyboard proposed by Gopher and Raij [31].

The chording keyboard used in this study has five keys, placed directly under the natural position of the fingertips. Unlike the Braille keyboard, it is designed to be operated by only one hand. In comparison to some of the devices presented above, with our device the users do not have to move their fingers from one key to another, so it should make no difference if they are able to see the keys or not. Considering this, we will only evaluate different feedback conditions regarding the typed text. Also, the small number of keys allows for higher design flexibility. The five keys can be directly integrated into a mobile phone case, around a computer mouse, or on a bike handlebar, thus being considered an extension rather than a separate object. This is important because some users might find it inconvenient to carry too many different devices.

III. Character Mapping

An important aspect of designing a chording keyboard is the mapping between the key combinations and the characters. One possibility is to assign easier combinations for more frequent letters, as in the Morse code, thus leading to higher typing speeds. Even if these mappings are easy to determine, the user must learn by heart the key-to-letter correspondence as there is no intuitive link between them. Another possibility is to use a semantically richer mapping, which would be easier to learn. The chording keyboard described in this paper is intended to be used in situations where desktop or other mobile keyboards are not appropriate, such as mobile environments. This is why we expect it to be used to type short texts, or to control a mobile device. Considering this, being able to easily learn the mapping is more important than being able to type fast.

We have designed the key-to-character mapping presented in this work with the primary goal of making it easy to remember. It is designed for a five-key keyboard, where each character is represented by a different key combination. From here on, we will focus only on lowercase letters, plus the period, space and backspace, as they are the most used. An additional button can be used to toggle between modes that enable typing uppercase letters, numbers, or other characters. The complete key map is given in the appendix.

To create enough possibilities for assigning an intuitive key combination to each character, we conceived five mnemonic categories. With them, a user usually can remember most combinations within minutes.

1) Single-key category: "**t**", "**i**", "**.**", "**r**", "**p**". Remembering the map for the characters in this category is totally trivial. Characters are produced by pressing a single finger and the letter is the initial of the finger. So, by pressing the key under **t**humb, **i**ndex, **r**ing and **p**inky, we obtain "**t**", "**i**", "**r**", and "**p**", respectively. There is an exception to the rule: since "**m**" fits well in another category (see below), we have reserved the middle finger for the period. The mnemonics for the characters in this category are presented in Figure 1.

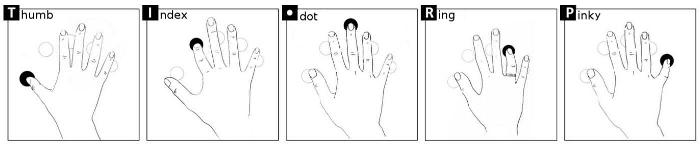


Figure 1. Single-key category

2) Fingers-down category: "**c**", "**m**", "**n**", "**u**", "**y**". The most natural way to produce the shape of a "**m**" with the hand is to stretch down the index, middle, and ring fingers, as shown in Figure 2. In a similar fashion, the shape of the fingers pressing the keys suggests the other letters in this category, namely "**c**", "**n**", "**u**", and "**y**".

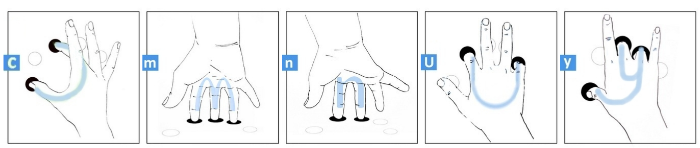


Figure 2. Fingers-down category

3) Fingers-up category: "**e**", "**l**", "**j**", "**v**", "**w**", **space**, **backspace**, **enter**. The idea is basically the same as for the previous category, but here we look at the fingers that are *not* used. A natural way to produce the shape of a "**w**" is to stretch up the index, middle, and ring fingers. The associated character is obtained by pressing the key(s) under the remaining fingers, as shown in Figure 3. "**v**", "**l**", "**e**", and "**j**", follow the same idea. We have included **space** and **backspace** in this category as **backspace** can be associated with the thumb pointing to the right and **space** with the pinky pointing to the left. For **enter**, the unused fingers represent a ∨ (pointing down) and a left arrow, suggesting the beginning of a new line.

4) Character landmark category: "**a**", "**f**", "**h**", "**k**", "**o**", "**s**", "**x**", "**z**". By looking at the shape of "**h**", we notice three landmark spots, and naturally enough, we associate them to the thumb, index and pinkie

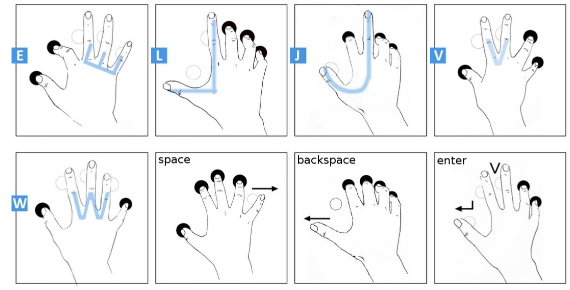


Figure 3. Fingers-up category

(see Figure 4). As a general rule, the thumb is for spots that are left and low, the index for left high, the ring for right high, and the pinky for right low. With a little bit of imagination, we can fit in this category also "**a**", "**f**", "**k**", "**o**", "**s**", "**x**", and "**z**". For "**o**", we imagine five dots spread around a circle, and we obtain it by pressing all buttons. For "**s**", we choose the points so that they remind us of a slalom (fingers-down and fingers-up alternate).

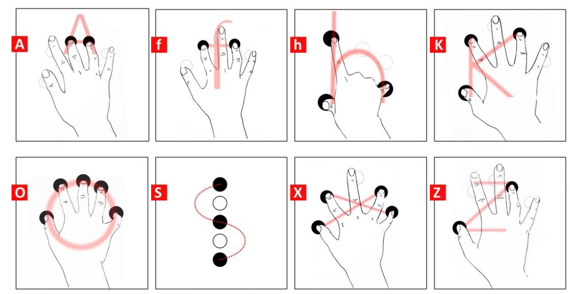


Figure 4. Character landmark category

5) Associative category: "**b**", "**d**", "**g**", "**q**". We remember these letters by associating them to similar letters. "**b**" and "**d**" can be seen as an "**o**" with a vertical bar on the left and right, respectively. We use the index and the ring fingers to represent these bars. "**g**" was inspired from "**y**" (they look alike in handwriting), and "**g**" inspires "**q**" (the tail ends left and right, respectively, so for "**g**" we use the thumb and for "**q**" the pinky). These mappings are shown in Figure 5.

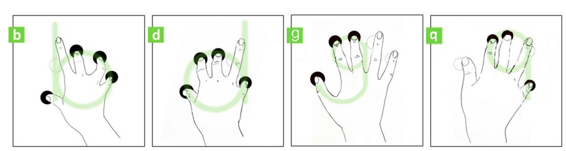


Figure 5. Associative category

The reader has probably noticed that some of the above mnemonics are easier to remember than others. With five keys, however, there are only 31 usable combinations and we use them all to map the 26 characters, plus the space, backspace, period, enter and comma. Hence, any change aimed at improving one mnemonic implies at least one other change.

The effectiveness of the proposed mapping is assessed through the studies described in the next sections. In the first study, we compare this mapping to two others from a learnability point of view, or how easy users can remember the key-to-character correspondences. In the second study, we estimate the usability of the mapping (typing speed, accuracy [32] and the most common mistakes), and how three different types of feedback affect the typing process.

IV. Learnability Study

This first study compares, from a learnability point of view, the proposed mapping (5keys) to two others. The references are the Microwriter mapping [15], also based on intuitive mnemonics, and the Baudot code [12], which is based on letter frequency and assigns easier key combinations to the most common characters. All three mappings are designed for five-key keyboards.

A. Experimental Setup

Design: The experiment had a 3×10 between-subjects design. Each of the 30 participants was assigned to work under one of the three conditions: 5keys, Microwriter or Baudot.

The experiment consisted of three sessions of three rounds each. For each participant, the sessions took place on consecutive days. For each round, the subjects had 5 minutes to look at a printed version of the mappings and try to remember them. Afterwards, they used a Java application to warm up, by typing each letter of the alphabet. During the warm-up phase, a help image showing the key combination for the letter to be typed was shown to the participants. In the next step, the help image was not available any more and the participants had to type the alphabet three times. The order of the letters was random, but the same for all participants. The subjects had five seconds and only one attempt to type each target character. The correct key combination was displayed for one second when the user typed a character (right or wrong), or when the user typed nothing for five seconds. The typing rounds were separated by breaks of two minutes.

Participants: We have recruited 30 participants, 10 for each of the three mappings, from the students of our university (undergraduate, master's and PhD programs). The participants were between 19 and 30 years old, and four were female. None of the subjects had used a chording keyboard before. As the participants who know how to play a musical instrument could have had an advantage, they were equally distributed among the three experiment groups. We also tried to equally distribute them based on gender and study level. Two participants abandoned the experiment after the first session, one testing the 5keys and one the Microwriter mapping.

Equipment and Software: We designed a Java application to simulate the chording keyboard on a regular QWERTY Apple desktop keyboard. It only allows for the use of five keys, each representing a key of the chording keyboard. Each of these keys corresponds to a finger of the right hand. A screenshot of the application is visible in Figure 6. The top-left window contains the target characters to be typed. The bottom-left window represents the typing area, and the help image is displayed on the right.

The Java application recorded log files containing the time of each key press and release, the typed text, the corresponding

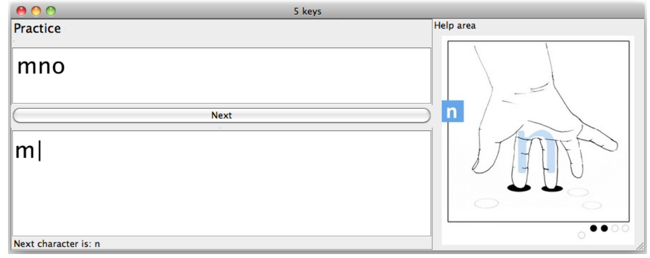


Figure 6. Application interface used during the study

key combination, the total number of errors and the total time spent writing each character.

Procedure: The participants were given written and verbal instructions regarding the goal of the experiment. The participants were given unique anonymous ID and were shown only the mapping that they were going to use. For each session of the experiment (approximately 30 minutes), they received a fixed monetary compensation.

During the experiment, the participants sat at a desk. Before the first session, the participants were explained how to press multiple keys to generate the chords, and were allowed to choose which five keys of the desktop keyboard they wanted to use. The only constraint was the space key for the thumb. They were not able to change the keys afterwards. A typical choice was the keys for f, t, y and u for the index, middle finger, ring and pinky, respectively.

The software was self-administered. Once started, it launches the warm-up phase, and then it goes automatically to the typing phase. The characters to be typed are also updated automatically.

B. Experiment Results

For each typing round, the participants had to type a total of 78 characters (3×26). To determine which of the mappings is easier to learn, we compared the number of errors (wrongly typed or not typed characters) for each round. Exponential regressions were derived to fit these error values. The average values for each mapping and for each round, and the exponential regressions are presented in Figure 7.

After two sessions (six rounds of approximately five minutes of typing each), the total number of errors was considerably lower for the mnemonic based mappings (5keys and Microwriter) compared to the mapping based on letter frequency. Therefore, we conclude that mnemonic based mappings are learned faster. This is confirmed by the anova test ($F = 24.15$, $p = 0.0001$). The goal of the study was to evaluate which mapping is easier to learn, and the Baudot mapping is clearly more difficult. Hence, in the third session, we only analyzed the 5keys and Microwriter mappings. Upon checking the average number of errors, no significant difference between these two was noticed ($F = 0.95$, $p = 0.358$). An advantage of 5keys can be observed from the analysis of the regression curves, because the curve for 5keys is slightly below the curve for Microwriter.

In addition to the total number of errors, the number of characters that were typed wrong at least once for each round

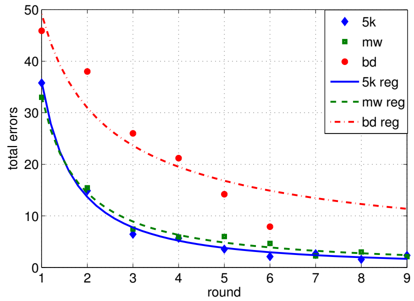


Figure 7. Average number of errors (for each mapping and for each round) and regression curves

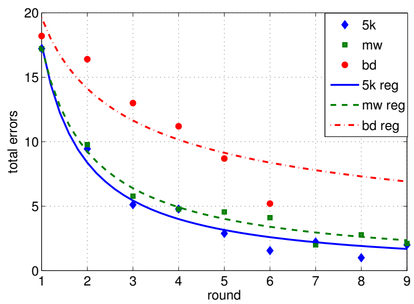


Figure 8. Average number of character errors (for each mapping and for each round) and regression curves

were also compared. For example, if "**a**" was typed incorrectly two times, this counted only as one character error. The results are shown in Figure 8 and provide an indication of how many characters are difficult to remember for each mapping. In this case, the difference between the 5keys and Microwriter mappings is more visible than in Figure 7, and statistically significant ($F = 5.4$, $p = 0.0486$). As expected, both mnemonic-based mappings lead to significantly less character errors than the Baudot mapping ($F = 18.65$, $p = 0.0004$). The Baudot mapping might lead to higher typing rates, but ascertaining this was not the goal of the presented study.

At the end of the third typing session (after nine rounds or approximately 45 minutes of actual typing), the participants were asked how confident they felt about their knowledge of the mappings and if they could use the presented method as a text input mechanism. All of them answered affirmatively and most mentioned that they had completely learned the mappings. This is confirmed by a low error rate (3.16% after 6 rounds and 2.14% after 9 rounds for the proposed mapping).

From this experiment, we draw the conclusion that a mnemonic-based mapping facilitates the process of learning the code. We also conclude that the proposed 5keys mapping outperforms the Microwriter mapping, also mnemonic-based, in terms of average error rate.

The mnemonic set was designed based on the finger positions of the right hand. Two of the participants (one for the 5keys and one for the Microwriter) were left-handed. Yet, they also typed with their right hand and, interestingly, their error rates were actually lower than the average.

V. Usability Study

The first study showed that an intuitive mapping can be learned in less than 45 minutes. It was followed by an independent experiment aimed at determining achievable typing rates, accuracy, and common error patterns for the 5keys mapping. Moreover, we evaluated different feedback types, when the text can and cannot be seen.

The input method that we present is designed to be used in situations where the visual attention is partially or totally unavailable for the typing process. In these conditions, audio feedback is often suggested as an alternative. This is indeed useful in some environments, but could be difficult to use in noisy areas. Considering this, we designed a 3×10 within-subjects experiment where we analyzed three different typing conditions. Under the first condition, subjects were able to see the outcome of what they have typed, under the second condition they received voice output for each typed letter (without visual feedback), and under the third condition they received no feedback at all about the typing. From here on, we will refer to these conditions as visual, auditive, and no-feedback, respectively.

A. Experimental Setup

Although there are a few similar aspects between this experiment and the previous one (same mapping and similar software interface), the study structure is completely different. Therefore, in this subsection, we will describe the new experimental setup.

Design: The experiment had a 3×10 within-subjects design. Each of the ten participants was asked to type under all three conditions.

The experiment was based on a Java application similar to the one shown in Figure 6, but the subjects were asked to type full sentences and used a chording keyboard prototype, not a desktop keyboard. The participants were asked to type for 10 sessions of 30 minutes. Each session consisted of three rounds of 10 minutes separated by breaks of 2 minutes, and each round corresponded to a different typing condition. The order of the typing conditions was random for each session, but the same for all subjects. For each user, the typing sessions took place on consecutive days, with the exception of weekends.

The first session enabled the subjects to remember the mapping between keys and characters. A help image showing the key combination for the letter to be typed was always displayed. During the subsequent sessions, the help image was only available on demand by pressing the *shift* key. At the beginning of each round, the participants warmed up by typing each letter of the alphabet. Afterwards, they typed phrases from a set considered representative of the English language [33]. These phrases were pre-prepared before the experiment to contain only small letters and no punctuation signs.

Participants: Ten participants took part in this study. Six of them also took part in the learnability study described in Section IV, using the 5keys mapping. The other four participants did the same experiment on a different occasion. Overall, they had approximately 45 minutes of training. All were PhD students from our university, eight male, two female, between 24 and 31 years old.

Equipment and Software: The keyboard prototype has the keys placed around a computer mouse and is presented in Figure 9. We designed the prototype in this way because we wanted the subjects to see a practical application of a chording device: allowing typing and screen navigation at the same time, with only one hand. The buttons are placed so that they can be easily operated while holding the mouse with the palm. We used keys and not pressure or touch sensors because they provide a distinct tactile feedback. The keyboard is designed using an Arduino Pro Mini microcontroller board and communicates with the computer by Bluetooth. The buttons are placed in a position that is naturally under the fingertips when the users hold their palm on the mouse.

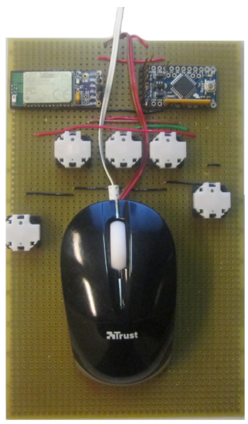


Figure 9. Chording keyboard prototype

The Java application is similar to the one described in Section IV. The typed text was displayed only for the visual condition. For the auditive condition, the participants used headphones to receive feedback. Log files containing the time of each key press and release, the typed text, the number of occurrences for each character, the corresponding key combination, the total number of errors and the total time spent writing each character were recorded. For each typing error, we checked what character was typed in lieu of the correct one.

Procedure: As before, the participants were given written and verbal instructions regarding the goal of the experiment. During the experiment, they sat at a desk, and each of them had an unique anonymous ID. The participants were instructed to type as quickly and as accurately as possible. Once a target text was completed, they were instructed to press the 'Next' button in order to display the next target text. They were told to not correct eventual mistakes and to keep typing, but this was not enforced and they were allowed to delete typed text. As a reward for the time commitment during the experiment, they received a fixed monetary compensation for the first nine sessions. For additional motivation, for the last session, the reward was proportional to the number of typed words and to the accuracy.

Experimental Data: The total amount of data gathered during the experiment consists of $40\,345$ words, out of which 4052 (10.17%) contain errors. The total number of characters is $219\,308$, from which 6386 (2.91%) are errors.

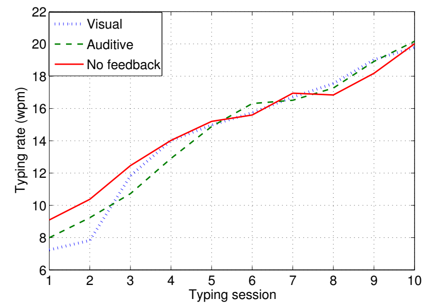


Figure 10. Average typing rates for each condition and for each typing session

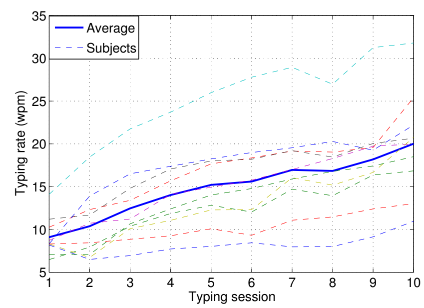


Figure 11. Average typing rates for each subject and for each typing session, for the no-feedback condition

B. Text-Entry Speed

We use the words-per-minute measure to describe the text-entry speed. This is defined as

$$wpm = \frac{60L}{t}\frac{1}{5}, \qquad (1)$$

where L is the total number of typed characters and t is the typing time in seconds. The scaling factor of 1/5 is based on the fact that the average English word length is approximately 5 characters. Because the average word length for the typed text differed from one session to another, the use of the above formula provides a more reliable estimate than actually counting the words.

In Figure 10, we show the average typing rates for each session and for each condition. For the first three sessions, the rates are higher for the no-feedback condition, and the anova tests showed that the differences are statistically significant ($F = 10.85$, $p < 0.0001$). From the fourth session onward, the differences between the typing rates are not so visible. Moreover, the effect of the feedback type is no longer significant ($F = 0.28$, $p = 0.75$). This probably happened because, in the beginning, subjects paused while typing to check the provided feedback, visual or audio. As they gained experience, they became more confident and did not analyze the feedback as often, therefore reducing the differences between conditions.

In Figure 11, we show the typing rates for each user and for each session, during the no-feedback condition. We notice that the fastest subject typed three times faster than the slowest subject, the differences being statistically significant ($F = 53.8$, $p < 0.0001$).

Figure 12. Chording keyboard prototype mounted on a bike handlebar. The four visible buttons correspond to the index, middle, ring and pinky fingers. The thumb button is on the other side of the prototype.

At the end of the experiment, the average typing rates were 19.77, 20.16 and 20.00 wpm for the visual, audio and no-feedback conditions, with maximums of 31.24, 30.48 and 31.78 wpm, respectively. Considering the participants' experience from the previous experiment, these values correspond to approximately 350 minutes of practice. Because the text entry rates would probably still improve, we use exponential regressions to estimate how fast people will be able to type after longer training periods. Based on these calculations, after 20 sessions (300 more minutes of practice), the average could be 26 wpm, and the fastest typist could reach 42 wpm.

As a reference, the typing rates achieved after 350 minutes of practice are 13.5 wpm for multi-tap mobile phones [5] and 24.2 wpm for Twiddler [19]. Rates of 20.36 wpm were reached by expert T9 users [34]. Handwriting speed is usually between 15 and 25 wpm [35]. We point out that the experimental conditions were not the same for all devices, hence, the above typing rates are only of indicative nature. For both multi-tap and T9 techniques, visual attention is essential for most users. For the 5keys device, it makes essentially no difference if the user has visual contact with the keys or not. It should also be taken into consideration that Twiddler uses 12 keys, whereas our mapping only requires 5 keys, thus providing a clear space advantage and more design flexibility. If placed in a position that is naturally under the fingertips (for example on the handlebar of a bike, as in the prototype from Figure 12), the users will have continuous access to the keys. Moreover, users do not have to move their fingers from one key to another, which probably leads to fewer errors.

C. Error Analysis

We used the total error metric presented by Soukoreff and Mackenzie in [36]: it considers both corrected and uncorrected errors. It is defined as

$$ErrorRate = \frac{IF + INF}{C + IF + INF} \times 100\%\,, \tag{2}$$

where C is the number of correctly typed characters, INF is the number of incorrectly spelled characters that were not corrected (it includes substitutions, when one character is typed for another, insertions, when an extra character is typed, and deletions, when a character is omitted), and IF is the number

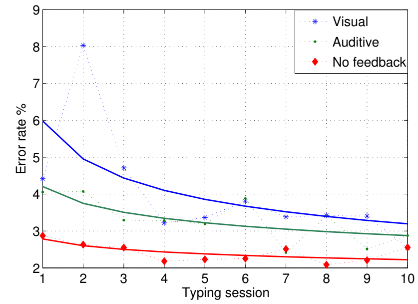


Figure 13. Average error rates and regressions for each condition and for each typing session

of incorrectly spelled characters that were corrected by the user.

The errors could have two main causes: the subject does not recall the correct key combination or, alternatively, a coordination mistake is produced during execution. We call these error types cognitive and sensorimotor errors, respectively. We expect the cognitive errors to decrease faster, as a function of training, because it is easier to learn the code than to improve motor skills. This is confirmed by the statements of the participants in the two experiments: they said that they had learned the mapping by the end of the training, and the errors were due to lack of attention or finger combinations that seemed more difficult.

In Figure 13, we display the average error rates for each session, accounting for both uncorrected and corrected errors, and the corresponding exponential regressions. All of the error rates are below 5%, except for the second typing session, visual condition. The reason for this could be the fact that in the first session the help image was always displayed, whereas in the second session it was hidden. Moreover, the first typing condition in session 2 was the visual one, giving subjects more practice time for the auditive and no-feedback conditions, which do not have much of an increase in the error rates. The averages for all sessions and for all users under the visual/auditive/no-feedback conditions are 3.41%, 2.97% and 2.32%, respectively. Anova tests show that feedback plays a relevant role in the error rates ($F = 25.57$, $p < 0.0001$).

Initially, it might seem surprising that the error rates are the lowest for the no-feedback condition and the highest for the visual condition. This is explained, however, by the fact that increased cognitive loads generally lead to more errors [37]. For our study, the cognitive load is the highest in the visual condition: users can check the whole typed phrase; it is reduced by the audio condition when users only hear the last typed character, and minimum in the absence of feedback. Noticing an error could cause someone to become less focused, thus favoring new mistakes.

The error rates decrease during the first four sessions (with the exception mentioned above), but afterwards they remain stationary or even increase. Similar effects, when after a certain point the error rates do not decrease anymore as users gain experience, are also noticed by Matias et al. [30] and Lyons et al. [19].

As in Section V-B, we compare the error rates with those

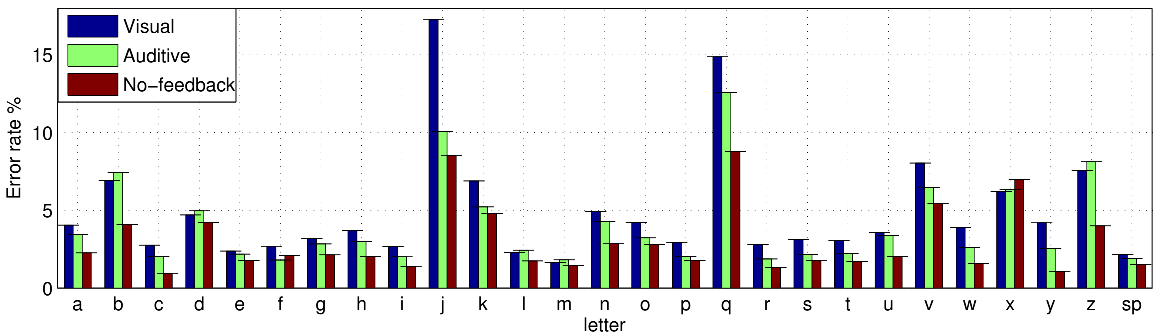


Figure 14. Error rates for each character for each typing condition

TABLE I. AVERAGE ERROR RATES(%) VERSUS THE REQUIRED NUMBER OF KEYS

Typing condition	Number of keys					Anova test
	1	2	3	4	5	
All conditions	2.06	3.01	4.99	4.73	3.40	$F = 1.15$, $p = 0.358$
Visual	2.74	3.89	6.39	5.00	4.20	$F = 0.92$, $p = 0.472$
Auditive	1.91	3.18	4.87	5.14	3.22	$F = 1.24$, $p = 0.323$
No-feedback	1.50	2.14	3.67	4.19	2.81	$F = 1.46$, $p = 0.248$

TABLE II. AVERAGE ERROR RATES(%) VERSUS THE MNEMONIC CATEGORY. SK - SINGLE-KEY; FD - FINGERS-DOWN; FU - FINGERS-UP; CL - CHARACTER LANDMARK; A - ASSOCIATIVE.

Typing condition	Mnemonic category					Anova test
	SK	FU	FD	CL	A	
All conditions	2.06	2.53	4.51	3.63	6.31	$F = 1.15$, $p = 0.358$
Visual	2.74	3.29	5.83	4.77	7.43	$F = 0.92$, $p = 0.472$
Auditive	1.91	2.72	4.21	3.91	6.94	$F = 1.24$, $p = 0.323$
No-feedback	1.50	1.63	3.28	3.30	4.80	$F = 1.46$, $p = 0.248$

for multi-tap (∼5%) and Twiddler (4.2%) after 350 minutes of practice, and also with expert T9 users (0.52%). Even if the Twiddler allows for higher typing rates, the error rates are also higher. Again, these values are only indicative, due to different experimental conditions.

It is important to analyze the error rates for each character, because this knowledge could be used to design a more efficient learning technique for the proposed mapping. For instance, subjects could be asked to practice more on characters with higher error rates. In Figure 14, we present the error rates for each character and for each typing condition. We notice that the character errors respect the pattern of the overall error rates: the highest for the visual condition and the lowest for the no-feedback condition: this is the case for 20 of the 27 analyzed characters.

For all three conditions, the error rates are higher for characters that are less frequent in the English language, such as "**q**" and "**j**", probably because the subjects had fewer opportunities to practice on them. Non-negligible error rates can also be observed for high-frequency characters, as users probably try to type faster as they gain more experience. The character error rates are similar between the three conditions, up to a scaling factor: if a character has an error rate lower than other characters for a specific condition, it usually also has a lower error rate relative to the same other characters for the other conditions. This is confirmed by the correlation coefficients between the error vectors, all above 0.9.

TABLE III. MOST FREQUENT LETTER SUBSTITUTIONS

substitution	percentage	substitution	percentage
$v \to b$	3.21	$n \to a$	1.88
$q \to j$	2.90	$q \to p$	1.69
$q \to d$	2.90	$j \to f$	1.61
$b \to y$	2.46	$x \to o$	1.49
$j \to x$	2.42	$d \to g$	1.42
$j \to q$	2.42	$v \to z$	1.35
$k \to h$	2.24	$p \to r$	1.34

In general, characters requiring three or four keys have higher error rates, but this is not statistically significant, as shown in Table I. Letters from the single-key category have the lowest error rates and those from the associative category the highest, but again, these results are not statistically significant (Table II).

D. Common Errors

To understand the error patterns that appear most frequently, we computed the confusion matrix [38] corresponding to the typed text. This is a square matrix with rows and columns labeled with all possible characters. The value at position *ij* shows the frequency of character *j* being typed when *i* was intended. The values are given as percentages from the total number of occurrences for character *i*.

In Table III, we present the 14 most common substitutions and the corresponding percentages. The values correspond to the whole experiment, including all three typing conditions. The confusion matrices for the whole experiment and for each condition are similar, with correlation coefficients higher than 0.99. If we consider only the erroneously typed characters (by setting the diagonal values, which are at least two orders of magnitude higher than the other values, to zero), the correlation coefficients are above 0.9, still showing a strong similarity: if one character is frequently typed instead of another under one condition, the same will happen under the other two conditions; if the probability for one character to be typed instead of another is low under one condition, it is also low under the other two conditions.

It is useful to represent a key combination by a 5-bit codeword in which the first digit represents the key under the thumb, the second digit the key under the index, etc. The value of a position is 1 if the corresponding key is pressed. So, for instance, 10111 is the codeword for "**b**", for which all fingers, except the index, press the keys. By analyzing the 5-

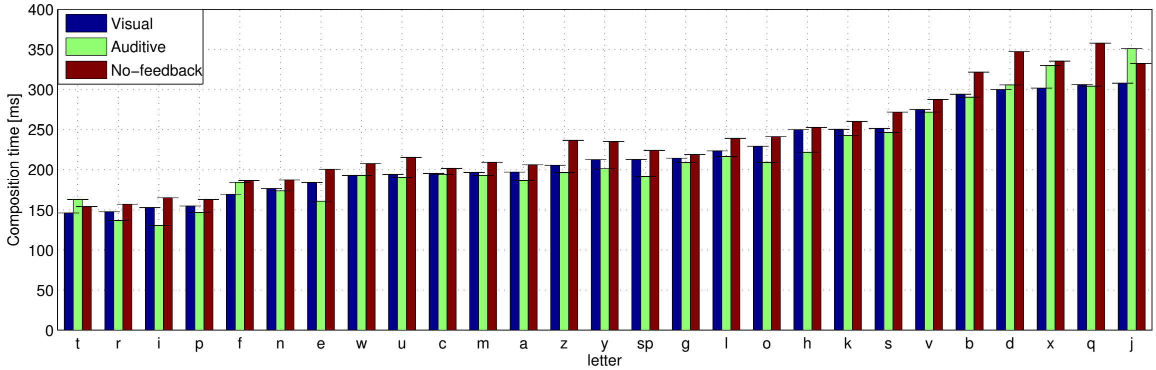


Figure 15. Composition times for each character for each typing condition

bit code for the substitutions from Table III, we notice that in 8 of the 14 cases the errors appear between characters that differ only by one bit (for example "**b**", code 10111, and "**v**", code 10011). In the other six cases, the errors appear between characters that differ by two bits. Overall, 44.81% of the wrongly typed characters differ from the intended character by one bit, 39.22% by two bits, 8.72% by three bits, 5.87% by four bits and 1.39% by five bits.

If we check word by word and consider only substitution errors, i.e., errors that arise from substituting individual characters (75% of the total errors), 91.48% of the erroneous words contain one substitution, 7.68% contain two substitutions and 0.67% three substitutions. From a bit-error point of view, 40.56% of the erroneous words contain a one-bit error, 40.92% a two-bit error, 7.62% a three-bit error, 7.20% a four-bit error and 2.13% a five-bit error. These values were used to implement an error correcting mechanism that relies both on a dictionary and on the probability that a character be substituted for another [39].

E. Character Typing Duration

As the coordination effort is not the same for all key combinations, we expect that different characters require more time than others to be typed. The time needed to form a key combination, called composition time, is measured from the moment the first key of a combination is pressed until a key is released. It is when a key of the combination is released that the corresponding character is produced. From that moment on, the pressing of a key indicates the start of a new character. In Figure 15, we present the average composition time for each character. Instead of ordering the characters alphabetically, we order them with respect to the composition time under the visual condition.

We notice that the composition times are higher for characters which are less frequent in the English language, such as "**q**" and "**j**", or for characters that require four keys, as "**x**", "**d**" and "**b**". Not surprisingly, "**o**" requires less time than the four-key letters, as it is easier to press all five keys than to press a specific subset of them. Also, letters requiring key combinations perceived as more difficult (for example "**q**", code 01101, or "**d**", code 11101, for which the middle finger and the pinky are down while the ring finger is up) require more time than others. In general, the

TABLE IV. AVERAGE COMPOSITION TIMES EXPRESSED IN MILLISECONDS VERSUS THE REQUIRED NUMBER OF KEYS

Typing condition	Number of keys					Anova test
	1	2	3	4	5	
All conditions	151	194	254	288	277	$F = 10.54$, $p < 0.0001$
Visual	150	189	248	277	229	$F = 13.31$, $p < 0.0001$
Auditive	144	184	245	279	209	$F = 8.25$, $p = 0.0003$
No-feedback	159	205	266	307	241	$F = 10.35$, $p < 0.0001$

TABLE V. AVERAGE COMPOSITION TIMES EXPRESSED IN MILLISECONDS VERSUS THE MNEMONIC CATEGORY. SK - SINGLE-KEY; FD - FINGERS-DOWN; FU - FINGERS-UP; CL - CHARACTER LANDMARK; A - ASSOCIATIVE

Typing condition	Mnemonic category					Anova test
	SK	FU	FD	CL	A	
All conditions	151	198	237	236	290	$F = 6.45$, $p = 0.0014$
Visual	150	195	232	231	278	$F = 7.01$, $p = 0.0008$
Auditive	144	190	230	227	277	$F = 4.77$, $p = 0.0064$
No-feedback	159	209	248	248	311	$F = 6.96$, $p = 0.0009$

composition time increases with the number of required keys per character, and the dependence is statistically significant. This is shown in Table IV. We also studied the dependency between composition times and the letter category and we summarize the results in Table V. These values confirm those from Table IV, as the average number of keys per character is 1 for the single-key category, 2.4 for fingers-down, 2.8 for fingers-up, 3 for character landmark, and 3.5 for the associative category. These results are also statistically significant.

As subjects gained more experience, they were able to type faster and the average letter duration decreased from 265.9 milliseconds in the first session under the visual condition to 183.9 milliseconds in session 10, or by 29.5%. During the same period, the text entry rates increased from 7.23 to 19.78 wpm, or by 173.8%. The difference is explained by the fact that the idle time between the end of one character and the beginning of the next also decreased.

VI. CONCLUSION

In this paper, we have presented the results of a study aimed at evaluating a mapping for a chording input device. The overhead needed to learn the mapping was reduced by choosing easy-to-remember key combinations. A first experiment showed that the mapping was learned after less than 45 minutes of actual typing. Moreover, the total number of errors was considerably smaller than for a letter-frequency based mapping and slightly smaller than for another mnemonic based mapping.

A second experiment enabled us to determine achievable typing rates, error rates, and the effect of different types of feedback for a chording keyboard. The subjects were asked to type under three conditions: with visual feedback, with auditive feedback and with no feedback at all. Due to the keyboard design, whether the user can see the keys or not should not make any difference on the typing process — at the end of the experiment, participants confirmed that they did not look at the keys. Similarly, someone playing a saxophone does not look at the keys to be pressed.

After approximately 350 minutes of typing (taking into consideration the previous typing experience of the subjects), the average entry rates are approximately 20 wpm under all three conditions, with the maximums above 30 wpm. We conclude, therefore, that having visual, audio or no feedback has no influence on the typing speed with the presented chording keyboard. The average error rates are 2.32% under the no-feedback, 2.97% under the auditive and 3.41% under the visual conditions. This is explained by the fact that the cognitive loads are different under the three typing conditions: the highest under the visual and the lowest under the no-feedback condition. Hence, not seeing the typed text actually provides an advantage. The error patterns are similar between conditions, the characters with the highest error rates and the most common substitutions being the same. We also analyzed which characters are perceived as more difficult to type and the most common errors. This data was used to develop an error correction mechanism specifically designed for a chording keyboard using the proposed mapping.

During the experiments, the subjects sat at a desk. To go one step further, we designed, built and tested a prototype for a bike, shown in Figure 12. We fit the five keys under the natural position of the fingers on the handlebar. With the help of a wrapper application that captured the input text, we used the keyboard to control the operation of a smartphone: controlling the music player, writing a short note, or initiating a phone call, without touching or looking at the touchscreen. Two of the authors tested the device and felt that they could effortlessly ride and type while controlling the bike with both hands and staying focused on the road. Moreover, as the keys are directly under the fingers, we could also type accurately on a slightly bumpy road. Though encouraging, these results are only exploratory, and performing a more detailed study will be difficult due to legal issues related to the risk of accidents.

The presented text entry method is not designed to completely replace desktop or on-screen keyboards, but to be used in certain specific situations. The typing rates comparable to handwriting speed, the low error rate, and the fact that the lack of visual or audio feedback does not impede the typing process make it a valuable option for situations where a person is not able to continuously check the output. In addition, the keyboard can be used with only one hand. The small number of keys also represents an advantage from the size and design flexibility point of view. Even if so far we envisaged the chording keyboard as a means of typing in dynamic or busy environments, due to its advantages, it can also be successfully used in other areas: for example, it can facilitate text input for disabled users who can only use one hand, or for persons who are visually impaired.

APPENDIX

During the study, we have only focused on small letters and space. However, with the help of a mode button, we can also type capitals, numbers and other symbols. The complete mapping is given in Table VI. The numbers are given by translating the code from binary to decimal. The least significant bit is the left one, because it is probably easier to think of the thumb as "one" and index as "two" than of the pinky as "one" and ring as "two". For other symbols, we tried to provide logical correspondences, such as the same code for "**a**" and "@", for "**m**" from minus and "**-**", for "**u**" from underscore and "**_**", etc. Other mappings may be less intuitive, but it is virtually impossible to assign easy-to-remember correspondences to all characters. Several symbols repeat, and this could be used to accommodate other characters, such as letters with diacritics.

TABLE VI. COMPLETE MAPPING

Five bit code	Character			
	Mode 1	Mode 2	Mode 3	Mode 4
00110	a	A	∼	@
10111	b	B	]	}
10100	c	C	5	(
11101	d	D	)	)
11000	e	E	3	=
01010	f	F	”	:
11100	g	G	7	&
11001	h	H	>	>
01000	i	I	2	up
01011	j	J	:	$
11010	k	K	*	ˆ
00111	l	L	\	/
01110	m	M	;	-
01100	n	N	6	—
11111	o	O	0	*
00001	p	P	(	right
01101	q	Q	=	”
00010	r	R	8	down
10101	s	S	+	+
10000	t	T	1	left
01001	u	U	’	_
10011	v	V	tab	tab
10001	w	W	¡	¡
11011	x	X	[	{
10110	y	Y	@	#
10010	z	Z	9	%
11110	space			
01111	backspace			
00011	new line			
00100	.	?	4	’
00101	,	!	-	;

REFERENCES

[1] A. Tarniceriu, P. Dillenbourg, and B. Rimoldi, "The effect of feedback on chord typing," in The Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '13, Porto, Portugal, September 2013, pp. 69–74.

[2] A. Tarniceriu, P. Dillenbourg, and B. Rimoldi, "Single-handed typing with minimal eye commitment: A text-entry study," in The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '12, Barcelona, Spain, September 2012, pp. 117–122.

[3] K. Lyons and R. Catrambone, "Improving novice performance on the twiddler one-handed chording keyboard," in Proceedings of the International Forum on Applied Wearable Computing, IFAWC '05, 2005, pp. 145–160.

[4] J. Scott, S. Izadi, L. S. Rezai, D. Ruszkowski, X. Bi, and R. Balakrishnan, “Reartype: text entry using keys on the back of a device,” in Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, MobileHCI ’10. ACM, 2010, pp. 171–180.

[5] I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner, “Letterwise: prefix-based disambiguation for mobile text input,” in Proceedings of the 14th annual ACM symposium on User interface software and technology, ser. UIST ’01. Orlando, Florida, United States. ACM, 2001, pp. 111–120.

[6] D. Wigdor and R. Balakrishnan, “Tilttext: using tilt for text input to mobile phones,” in Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST ’03. ACM, 2003, pp. 81–90.

[7] URL: http://www.frogpad.com/ [accessed: 2014-05-12].

[8] D. Goldberg and C. Richardson, “Touch-typing with a stylus,” in Proceedings of the INTERACT ’93 and CHI ’93 conference on Human factors in computing systems, CHI ’93. ACM, 1993, pp. 80–87.

[9] J. O. Wobbrock, B. A. Myers, and J. A. Kembel, “Edgewrite: a stylus-based text entry method designed for high accuracy and stability of motion,” in Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST ’03. ACM, 2003, pp. 61–70.

[10] P. Isokoski and R. Raisamo, “Device independent text input: a rationale and an example,” in Proceedings of the working conference on Advanced visual interfaces, AVI ’00. New York, NY, USA: ACM, 2000, pp. 76–83.

[11] D. J. Ward, A. F. Blackwell, and D. J. C. MacKay, “Dasher a data entry interface using continuous gestures and language models,” in Proceedings of the 13th annual ACM symposium on User interface software and technology, ser. UIST ’00. New York, NY, USA: ACM, 2000, pp. 129–137.

[12] A. Ralston and E. D. Reilly, Eds., Encyclopedia of computer science (3rd ed.). Van Nostrand Reinhold Co., 1993.

[13] F. C. Bequaert and N. Rochester, “Teaching typing on a chord keyboard,” IBM Technical Report, Tech. Rep., 1977.

[14] D. C. Engelbart, “Design considerations for knowledge workshop terminals,” in Proceedings of the June 4-8, 1973, national computer conference and exposition, AFIPS ’73. ACM, 1973, pp. 221–227.

[15] URL: http://www.ericlindsay.com/palmtop/mwrite.htm/ [accessed: 2014-05-12].

[16] URL: http://www.xaphoon.com/dataegg/ [accessed: 2014-05-12].

[17] URL: http://gkos.com/ [accessed: 2014-05-12].

[18] URL: http://chordite.com/ [accessed: 2014-05-12].

[19] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney, “Twiddler typing: one-handed chording text entry for mobile phones,” in Proceedings of the SIGCHI conference on Human factors in computing systems, CHI ’04. Vienna, Austria: ACM, 2004, pp. 671–678.

[20] URL: http://www.ekatetra.com/products/ekapad.html [accessed: 2014-05-12].

[21] R. Rosenberg and M. Slater, “The chording glove: a glove-based text input device,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 29, no. 2, 1999, pp. 186 –191.

[22] K. Lyons, D. Plaisted, and T. Starner, “Expert chording text entry on the twiddler one-handed keyboard,” in Proceedings of the Eighth International Symposium on Wearable Computers, ISWC ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 94–101.

[23] F. E. Sandnes, Y.-P. Huang, and Y.-M. Huang, “Near eyes-free chauffeur computer interaction with chording and visual text mnemonics,” J. UCS, vol. 16, no. 10, 2010, pp. 1311–1326.

[24] F. E. Sandnes and Y.-P. Huang, “Chording with spatial mnemonics: Automatic error correction for eyes-free text entry,” J. Inf. Sci. Eng., vol. 22, no. 5, 2006, pp. 1015–1031.

[25] F. Sandnes and Y.-P. Huang, “Non-intrusive error-correction of text input chords: a language model approach,” in Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American, june 2005, pp. 373 – 378.

[26] M. Silfverberg, “Using mobile keypads with limited visual feedback: Implications to handheld and wearable devices,” in Human-Computer Interaction with Mobile Devices and Services, Lecture Notes in Computer Science, L. Chittaro, Ed. Springer Berlin Heidelberg, 2003, vol. 2795, pp. 76–90.

[27] J. Clawson, K. Lyons, T. Starner, and E. Clarkson, “The impacts of limited visual feedback on mobile text entry for the twiddler and mini-qwerty keyboards,” in Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on, oct. 2005, pp. 170 – 177.

[28] S. Mascetti, C. Bernareggi, and M. Belotti, “Typeinbraille: Quick eyes-free typing on smartphones,” in Computers Helping People with Special Needs, ser. Lecture Notes in Computer Science, K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds. Springer Berlin Heidelberg, 2012, vol. 7383, pp. 615–622.

[29] K. Yatani and K. N. Truong, “An evaluation of stylus-based text entry methods on handheld devices studied in different user mobility states,” Pervasive and Mobile Computing, vol. 5, no. 5, 2009, pp. 496 – 508.

[30] E. Matias, I. S. MacKenzie, and W. Buxton, “One-handed touch typing on a qwerty keyboard,” Hum.-Comput. Interact., vol. 11, no. 1, Mar. 1996, pp. 1–27.

[31] D. Gopher and D. Raij, “Typing with a two-hand chord keyboard: will the qwerty become obsolete?” Systems, Man and Cybernetics, IEEE Transactions on, vol. 18, no. 4, july-aug. 1988, pp. 601 –609.

[32] R. W. Soukoreff, “Text entry for mobile systems: Models, measures, and analyses for text entry research,” Master’s thesis, York University, 2002.

[33] I. S. Mackenzie and R. W. Soukoreff, “Phrase sets for evaluating text entry techniques,” in Extended Abstracts of the ACM Conference on Human Factors in Computing Systems CHI ’03. Fort Lauderdale, Florida, United States: ACM, 2003, pp. 766–767.

[34] C. L. James and K. M. Reischel, “Text input for mobile devices: comparing model prediction to actual performance,” in Proceedings of the SIGCHI conference on Human factors in computing systems, CHI ’01. Seattle, Washington, United States: ACM, 2001, pp. 365–371.

[35] I. S. MacKenzie and R. W. Soukoreff, “Text Entry for Mobile Computing: Models and Methods, Theory and Practice,” Human-computer Interaction, vol. 17, 2002, pp. 147–198.

[36] R. W. Soukoreff and I. S. MacKenzie, “Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’03. New York, NY, USA: ACM, 2003, pp. 113–120.

[37] A. Baddeley, Working Memory, Oxford Psychology Series, No 11. Clarendon Press, 1986.

[38] K. Kukich, “Techniques for automatically correcting words in text,” ACM Comput. Surv., vol. 24, December 1992, pp. 377–439.

[39] A. Tarniceriu, B. Rimoldi, and P. Dillenbourg, “Error correction mechanism for five-key chording keyboards,” in The 7th International Conference on Speech Technology and Human-Computer Dialogue, SpeD ’13, Cluj-Napoca, Romania, October 2013.

Applying a Layered Model for Knowledge Transfer to Business Process Modelling (BPM)

Felix Schiele, Fritz Laux
Fakultät Informatik
Reutlingen University
Reutlingen, Germany
Felix.Schiele@Reutlingen-University.de
Fritz.Laux@Reutlingen-University.de

Thomas M Connolly
School of Computing
University of the West of Scotland
Paisley, UK
Thomas.connolly@uws.ac.uk

***Abstract*— Knowledge transfer is very important to our knowledge-based society and many approaches have been proposed to describe this transfer. However, these approaches take a rather abstract view on knowledge transfer, which makes implementation difficult. In order to address this issue, we introduce a layered model for knowledge transfer that structures the individual steps of knowledge transfer in more detail. This paper gives a description of the process and also an example of the application of the layered model for knowledge transfer. The example is located in the area of business process modelling. Business processes contain the important knowledge describing the procedures of the company to produce products and services. Knowledge transfer is the fundamental basis in the modelling and usage of business processes, which makes it an interesting use case for the layered model for knowledge transfer.**

Keywords-knowledge transfer; knowledge conversion; impart knowledge; business process modelling.

I. Introduction

This paper presents an approach that describes the transfer of knowledge. It extends the original paper "A Layered Model for Knowledge Transfer" [1] with an example in the area of business process modelling. The application in the area of business process modelling pursues two goals. First, the example intends to make the mode of operation of the layered model for knowledge transfer apparent. Second, the application in the area of business process modelling is to show how the layered model can be used in solving real problems in modelling of business processes.

In our knowledge-based society, the relevance of knowledge transfer is increasing. Knowledge management and the understanding of economic coherency can help an organization to handle the challenges of an increasingly fast-evolving environment [2]. The transfer of knowledge from one person to another is of major importance for enterprises [3]. The Socialization, Externalization, Combination, and Internalization (SECI) Model of Nonaka and Takeuchi [4] is an approach that supports organizations in the handling of the important knowledge resource and describes knowledge conversions between internal and external knowledge. However, the SECI Model does not contain precise descriptions of knowledge transfer. This paper aims to introduce a model for knowledge transfer that makes problems emerging during the transfer visible and explainable, and facilitates its implementation through a more detailed and clearer structuring.

This paper is structured as follows: Section II discusses and provides working definitions of data, information and knowledge. Section III discusses existing communications models and Section IV proposes a model of knowledge transfer that aims to reduce errors on each of the knowledge levels. Section V describes an application of the layered model for knowledge transfer in the area of business processes. Section VI draws conclusions and discusses future directions.

II. Data, Information, Knowledge, Conversation, and Communication

As mentioned by Nonaka [5], the terms information and knowledge are sometimes used interchangeably even though they have different meanings. In her study on the wisdom hierarchy, Rowley [6] pointed out that it is especially important to define the concepts of data, information, and knowledge. Since this paper focuses on the transfer of knowledge, the following section presents definitions to distinguish the terms data, information and knowledge. Having examined various definitions the authors will present their own definitions, which are based on some of the previously introduced ones.

A. Data

Hasler Roumois [7] stated that data consist of symbols that are combined into words by using syntax. The words receive a semantic meaning when they are associated to things. Davenport and Prusak [8] describe data as the raw material for information without an intrinsic meaning. A data set can contain facts about an event or thing. This is also the view of Wormell cited in Boisot and Canals [9] that data are alphabetic or numeric signs that without context do not have any meaning. Rainer [10] characterized data items as "*an elementary description of things, events, activities, and transactions that are recorded, classified, and stored but are not organized to convey any specific meaning*." Ackoff [11] viewed data as "*symbols that represent properties of objects, events and their environment. They are products of observation*." Frické [12] criticized the opinion of those who say that data have to be true, which means that the statement of the data must be true. The following example confirms Frické's criticism: consider a data set containing incorrect or imprecise data, then according to the others this data would not be considered data. Weggeman [13] differentiates between hard and soft data. If the measuring technique and the measurement that created the data are unequivocal, Weggeman describes them as hard data, otherwise the data are softer. Weggeman's classification requires, however, knowledge about the data and the things they represent, which is beyond the scope of data, instead part of the scope of information.

1) Definition: data

Data consist of symbols that are combined into words by using syntax. Data are produced by humans or machines. They can be the result of observations of the real world, descriptions of abstract things, or the result of processing existing data. Data cannot be true or false since this decision is beyond the scope of data.

B. Information

In the definition of information, there are two fundamentally different theories. The more technical approach characterizes information as data where context has been added [14]. In the more philosophical approach it depends on the receiver whether something is information or only data. Hasler Roumois [7] stated that when people recognize the meaning of data and consider their relevance they become information. Similarly, Davis and Olson [15] view information as data that has been processed into a form that is meaningful to the recipient. Dretske [16] noted about information: "*Roughly speaking, information is that commodity capable of yielding knowledge, and what information a signal carries is what we can learn from it. If everything I say to you is false, then I have given you no information*". However, the recipient of the message may receive the meta information that the other person is lying, Dretske stated. Weggeman [13] provides the example that an author will look at his book as information whereas others may consider it initially as a collection of data. It is up to the receiver to consider whether the data are relevant or not. Weggeman argues that data becomes information even if it is irrelevant to the recipient, because the assessment is a form of recognition that leads to information. As stated in the example from Dretske, the recipient may receive meta information. For this analysis the receiver had to compare the message with his personal knowledge base. If he already knew the content, this may lead to reinforcement by the additional confirmation through the message. Therefore, the authors agree with Dretske that the receiver may achieve meta information, but in this case the data does not become information. Rainer and Cegielski [10] described information as organized data that have meaning and value to a recipient.

1) Definition: information

Data becomes information when a person receives data, decodes them, recognizes the meaning and considers them relevant. If the data do not contain anything new for the receiver, the data do not become information. However, they may result in meta information, such as confirmation of the known.

C. Knowledge

For the processing of information the existing knowledge is of crucial importance. Wormell, cited in Boisot and Canals [9], believes knowledge is enriched information by a person's or a system's own experience; it is cognitive based; it is not transferable, but through information we can communicate about it. Dretske represents the relation of information and knowledge as follows: "*Knowledge is identified with information-produced (or sustained) belief, but the information a person receives is relative to what he or she already knows about the possibilities at the source*" [16]. About knowledge Polanyi [17] said: "*I shall reconsider human knowledge from the fact that we can know more than we can tell*". Thus he shows that knowledge has a secret or tacit part and not everything a person knows can be passed. Polanyi describes explicit knowledge, which in turn can be expressed in formal, semantic language, and tacit knowledge, which is personalized and therefore hard to express [18]. According to Nonaka [19] explicit knowledge is knowledge that can be articulated into formal language, such as words, mathematical expressions, specifications and computer programmes, and can be readily transmitted to others. This is in contrast to tacit knowledge, which is personalised and based upon experience, context and the actions of an individual; tacit knowledge resides in individuals who may be unaware that they possess such knowledge. There is also implicit knowledge, which refers to knowledge that is revealed in task performance without any corresponding phenomenal awareness; implicit knowledge is often expressed unintentionally. This characteristic is described as type dimension of knowledge [20]. For this article, the explicit type of knowledge represents the most important knowledge type, because it is the knowledge that can be easily externalized. Weggeman [13] firmly believes that information and knowledge only exist inside the person whereas data can exist outside a person. Davenport and Prusak [8] describe knowledge as bound to a person: "*It [knowledge] originates and is applied in the mind of the knowers*." The transformation from information to knowledge takes place when the information is linked to the existing knowledge through a thinking process [7]. The authors propose the term knowledge base as the collection of all facts, rules, and values that are represented in the brain of a person. Spitzer [21] depicts that through the learning process links are created or dissolved in the brain, which results in changes of the knowledge base. Spitzer [21] points out, that messages, which have the quality of relevance and novelty, can be memorized easily.

1) Definition: knowledge

Information becomes knowledge if a thinking process occurs in which the information is linked to the existing knowledge and is stored persistently. The quality of information being relevant and new, insofar as there is a difference to the existing knowledge, encourages the permanent memorization of information. Based on the input by the information, the knowledge base of the person may be extended or restructured.

D. Knowledge Conversion

Nonaka and Takeuchi [4] described the conversation of knowledge in their SECI Model. For this work externalization and internalization of knowledge are of particular importance. Nonaka and Takeuchi describe the internalization as conversion from explicit to tacit knowledge and the externalization as conversion from tacit to explicit knowledge. The authors use the concepts of externalization and internalization with respect to the conversion of data to knowledge and vice versa. Externalization enables a person to converse parts of the personal knowledge base, making them accessible to others. For example, if someone writes down what he knows, everyone except him will refer to this as data. Internalization will happen when a reader receives new knowledge by reading and learning from it.

Transfer and persistent storage require an externalization of knowledge in a recognized and structured language. The various levels of messages are related to levels of semiotics, which are syntactic, semantic and pragmatic. Krcmar [22] states that syntax declares the rules according to which characters can be combined to words and these can be combined to sentences. The relation between words and objects represented by the words as the relationship between characters is denoted by semantics. The intention of a person sending words as a message is explained as pragmatic.

E. Communication

The protagonist of systems theory, Luhmann [23], explained communication as a process consisting of three steps of selection. In the first step, the sender decides which information he wants to pass on. In the second step, he selects a single message from many possible messages. In the last step, the recipient selects the information out of the message thereby completing the communication. Based on Luhmann's work, Berghaus [24] describes several results, which can occur if a sender is forwarding a message to a receiver.

- Case 1: The receiver picks up the message and interprets it in the desired way.
- Case 2: The receiver picks up the message but interprets it differently.
- Case 3: The receiver does not recognize the message as a message.

Only one of the three cases achieves the desired result. In this paper the second case and the various reasons for the error in communication will be considered in more detail. The third case plays a minor role as it is assumed that the message is detected as a message because only the messages presented as data are considered.

III. RELATED WORK: COMMUNICATION MODELS

A. Schema of Social Communication

Figure 1 shows Aufermann's [25] model for social communication in which two parties are involved. The sender encodes the statement he intends to submit in a message. Therefore, he uses his own character set to encode the message. The message is sent via a medium to the recipient whereby spatial and temporal distance is overcome. When receiving the message the recipient will use his own character set for the decoding of the message.

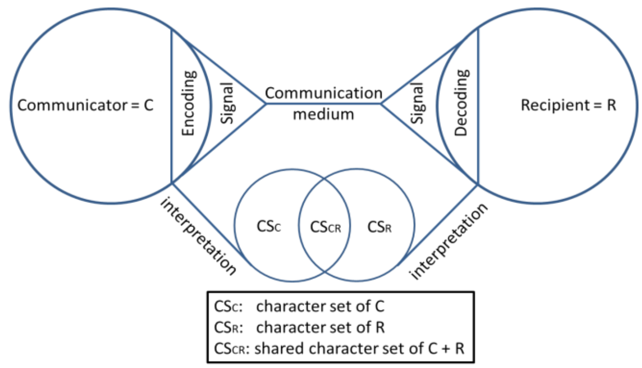


Figure 1. Schema of Social Communication [25] (German)

The model illustrates the important point of the character sets used by sender and recipient and the need to use only those characters that are within the shared character set.

B. A Mathematical Theory of Communication

In Shannon's description of the operation of a communication system, the sender is named "information source" and the receiver is called "destination" [26]. Shannon has investigated the frequency of characters contained in a message, and compared the expected and the actual occurrence of a character. Using the 'entropy' Shannon invented a key figure to measure the information contained in a message. Due to the technical use of the model, specifically the control of missiles, the emphasis is on the transmission of the signal [27]. In addition to Aufermann's schema of social communication, Shannon's model describes the influence of the transmission of a signal by a noise source.

C. Four Forms of Knowledge Conversion

The SECI Model, developed by Nonaka and Takeuchi [4] is focused on the knowledge conversions during knowledge transfer. The description of four conversions takes place at an abstract level showing the particularities of each conversion. However, a detailed description of the individual conversions is missing. Nonaka and Takeuchi describe socialization as a direct knowledge transfer from the tacit knowledge of one person to the tacit knowledge of another person, enabled by action and observation. However, this abstract view does not show exactly how knowledge is transferred in this case. A situation in which socialization happens may arise when master and apprentice work together. Even though the master does not express his knowledge intentionally he externalizes it through his action. Based on the perceived action and the results of action, the apprentice will unconsciously obtain knowledge by internalization.

D. A Hierarchical Modelling Approach to Intellectual Capital Development

Ammann [20] describes knowledge conversions from one person to another, in which the different types of knowledge are taken into account. In addition to the knowledge conversions described in the SECI Model the conversion from latent or conscious knowledge to explicit knowledge is described. Even though Ammann's approach represents knowledge transfer in greater detail, this approach does not give a precise description of how the transmission works.

IV. Model of Knowledge Transfer

A message is a possible way to impart knowledge. The correct interpretation of the message may be prevented by interferences that can affect the message. As described by Shannon the disruption may be caused by a noise source disturbing the medium transmitting the message. In addition to the interferences from the outside that may influence the transport medium, the personal knowledge base of the sender and the receiver may also affect the knowledge transfer. The influence of the transfer through the personal knowledge of sender and receiver can take place in four layers. The interpretation of the message depends on the elements that are used and whether they are part of the knowledge base of the receiver and equivalent to the elements of the sender's knowledge base.

A. Layers that Influence the Transfer

The four layers that influence the transfer of a message from one person to another are code, syntactic, semantic and pragmatic layer. The concept of a knowledge transfer through different layers was influenced by the OSI Reference Model [28]. Figure 2 illustrates the transfer of a message from the sender to the receiver passing through the four layers.

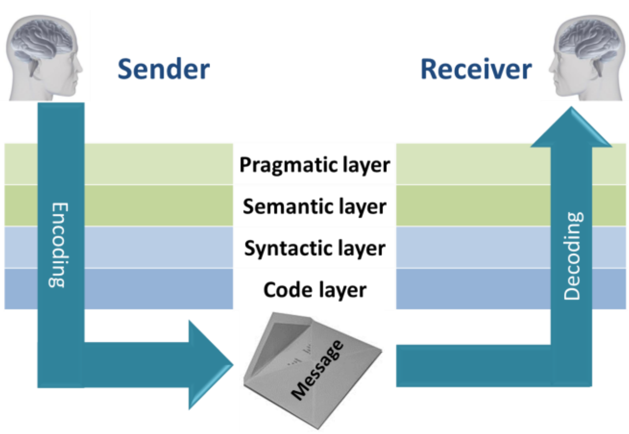


Figure 2. Knowledge Transfer through four layers

1) Code Layer

At the lowest level of the layer for transfer is the code. The code consists of symbols or signs that represent the smallest units, which form the basis of the higher layers. In the case of written language, which is the focus here, the smallest elements are the characters, σ, taken from an alphabet Σ. In the case of spoken language it would be phonemes, or in sign language gestures.

2) Syntactic Layer

The second layer is constituted by the syntax that contains rules for the combination of signs or symbols. In written language, L, the characters σ are combined to form words ω by the use of production rules P.

3) Semantic Layer

The third layer contains the semantics that establish the relation between words ω and meaning m. This relation, called semantics s(ω, m), connects the word to its meaning, which can be a real world entity or an abstract thing.

4) Pragmatic Layer

The top layer is the pragmatic layer. Pragmatics p(s, c) connects the semantic term s with a concept c. The concept contains the course of action and the aims and moral concepts that are represented in the human brain. They influence the thinking and acting of sender and receiver.

B. Process of a Knowledge Transfer via Messages

The aim of the following example is the desire of a person, called sender, to communicate something to another person, called receiver. Even if the model is general, the focus is on the written notification.

1) Sender: Pragmatic Layer

The core of the message is represented in the pragmatic layer. The aims and moral concepts of the sender do not only affect the externalization of the message, but also the assumptions he makes about the receiver.

2) Sender: Semantic Layer

This layer contains all words ω and their relation to the objects. The sender must choose appropriate words that are available in his personal knowledge base. *Appropriate* means not only the term that fits best, but also that refers to the knowledge of the recipient.

3) Sender: Syntactic Layer

This layer contains the rules P according to which the sentences and terms are made. The words ω chosen to carry the meaning are wrapped in sentences. Again, the sender must choose the words in compliance with the words known by the recipient.

4) Sender: Code Layer

To transfer the message as written communication the sender has to write the words ω by using characters σ that are part of an alphabet Σ of a language.

5) Transfer: Message

The communication medium (e.g., letter, email) transmits the data from the sender to the receiver.

6) Receiver: Code Layer

The receiver will view the message and read the characters σ, if he knows them. In the case where the message contains characters from an alphabet unknown to the receiver, the transfer might be disrupted. With only small deviations of the used characters a reconstruction might be possible, otherwise it can lead to misinterpretation or stop the decryption.

7) Receiver: Syntactic Layer

The receiver will compose the characters σ to words ω and sentences if they are part of a language L he knows. As in the decoding of the code small differences can be compensated under favourable circumstances, otherwise misinterpretation or stopping the decryption are the consequences.

8) Receiver: Semantic Layer

Almost simultaneously with the combination of words and sentences the receiver will put the terms in relation to the things for which they stand. The more the receiver knows the context and the sender of the message, the easier it is to capture the meaning of the text.

9) Receiver: Pragmatic Layer

In a final step the receiver will interpret the message in relation to his own aims and values. The things the receiver knows about the sender as well as the assumptions regarding the receiver that are influenced by the sender's own values and aims, play an important role in the decoding of the message.

C. Influence of Overlapping Knowledge

Knowledge about the receiver is an important requirement for a successful and lossless transfer of a message. The better the sender knows the receiver, the easier he can encode the message. A proper encoding of the message can be done by using elements that exist identically in the personal knowledge base of the sender as well as in the personal knowledge base of the receiver. If the receiver is unknown, only assumptions can be made to support the selection. The other way around it is easier for the receiver to decode the message if he knows the sender of the message very well. Figure 3 visualizes the overlapping of the knowledge in different layers.

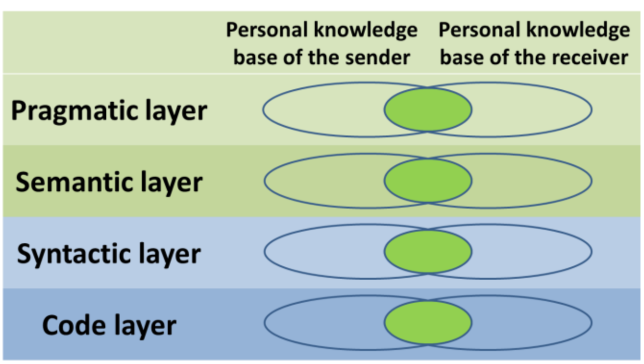


Figure 3. Overlapping Knowledge

D. Example of Knowledge Transfer

A challenge in knowledge transfer is the different knowledge base of sender and receiver. In companies, this situation may occur when a business analyst explains a modelled process to a technician in a department. The business analyst, an expert in business process modelling (BPM), will interview the employees of the department to review the department's processes. During the interview he will make notes and sketches, which he subsequently transfers to business process models.

The business analyst will show and explain the modelled processes to the departmental employees to check that everything has been modelled properly so that model and practised processes are consistent. When explaining the model to the technician, the business analyst must take into account that the technician might not have (sufficient) knowledge of a business process modelling language. We assume that the business analyst and the technician speak the same language and have had similar schooling. Consequently, symbols that exist in their knowledge base are nearly equal although the business analyst might know additional symbols such as those used in the business process modelling languages. This consensus also occurs in the syntactical layer, which contains rules to build words, and the semantic layer, where things are represented through words. The largest differences in the knowledge base are probably found in the pragmatic layer. The basic concepts of aim and moral that are shaped by education, culture, and environment, may be similar for both. However, the business analyst might have a larger knowledge base in the respective aims and concepts of BPM, while the technician might have a larger knowledge base in the respective aims, processes, and concepts of his special field.

The business analyst, after seeing that the technician has not mastered a business process modelling language, will avoid using terms and concepts unknown to the technician. When explaining the model, the business analyst will introduce the necessary symbols, terms, and concepts to explain the process. He can try to use simple explanations and he can bring in additional information that facilitates the interpretation of the message. The interpretation of the symbols is dependent on the knowledge base of the interpreting person. The interpretation can be facilitated by restrictions; in this example, the terms used for the process are terms from the domain of the department as well as from BPM. The context the terms are used in thereby facilitates the correct interpretation of the process.

E. Supporting the transfer of the message

The prevention of transmission errors as well as misinterpretation of the message is crucial for the knowledge transfer. The dissemination of knowledge in the form of written language uses characters to form the message. The message shall not only contain plain text but also data types. Data types are object descriptions, which define and cluster objects of the same kind. This implies that all objects of a type have the same value range and usability. This supports the accurate decoding of the message, because the knowledge about the data type restricts the room for interpretation. Prerequisite for the use of data types is that sender and receiver know the data types used in the message. Through the use of data types the message can be mapped corresponding to the layers of the model. Although the message can contain only data, a part of the logic necessary to extract the meaning, can be embedded by the use of data types.

V. Application in Business Process Modelling

We apply the model for knowledge transfer in the area of business processes. The important knowledge of a company, describing the procedures for the production of products and services, is incorporated in business processes. Due to this fact they are an interesting area of application. Furthermore, the modelling of business processes normally requires bidirectional communication. An exemplary application of the layered model for knowledge transfer is presented using the example of a business process for the procurement of goods. The business process for procurements is a process that is used in almost every producing company usually with small company-specific adjustments. Event-driven Process Chain (EPC) diagrams will provide a basis for the graphical representation of the business process.

A. Event-driven Process Chains

The EPC diagram was invented by Scheer [29] and consists of events and functions interconnected by a control flow. Functions are rectangular symbols that represent a performance to achieve a desired result. Events are hexangular symbols that represent a situation that triggers a function and they are used to represent the result of a function. The control flow can be split and merged by the use of a logical connector such as AND, OR, and XOR [30].

Another basic symbol used in the EPC is the role showing who is performing a function. If further symbols, standard or company-specific, are added to the EPC it becomes an enhanced Event-driven Process Chain (eEPC). Common used symbols are resources such as system, data base, and document, which can be connected to functions by an information flow indicating the input or output of a function. EPC is a semi-formal modelling language especially for the functional representation of business processes. EPCs are widespread, at least in Germany and supported by various tools [31].

B. Process: Procurement of Goods

Figure 4 shows the business process for the procurement of goods modelled as an EPC diagram. The process starts when the event "goods required" occurs. The event can be activated by a material requirement planning in an enterprise system such as SAP as well as by a manual requirement planning carried out by a human. The first activity of the business process is the function "perform order" where an order is created for the purpose of covering the demand. This process step, undertaken by a procurement manager, results in an order with an order number including all relevant details. At this point the process splits and is waiting until the events "goods arrive" and "bill arrives" occur. The AND symbol splitting the process is a logical AND which illustrates that both following events will occur. An important rule for modelling processes states that the logical connector that splits the process must also be used for merging.

Once the goods arrive a logistic employee will accept the goods and check the delivery in compliance with the order number. The result of the goods received is that the inventory of goods is increased by exactly the number of delivered goods. The invoice checking runs in parallel as soon as the bill arrives and triggers the event "bill arrives". Following this event the "invoice checking" is performed by a financial accountant. Here, the data of the invoice are matched with the expected data, which are based on the order. The data of the invoice are entered into the system. Here the process is merged again visualized by a logical AND symbol. When used for merging the process the AND requires all inputs to be true until the process can continue. As soon as both parallel process flows are completed a financial accountant performs the payment. Result of this function "perform payment" is that the supplier obtains the money and the company's bank account is decreased by the same amount. The process is completed by this cash transfer and ends with the event "order completed".

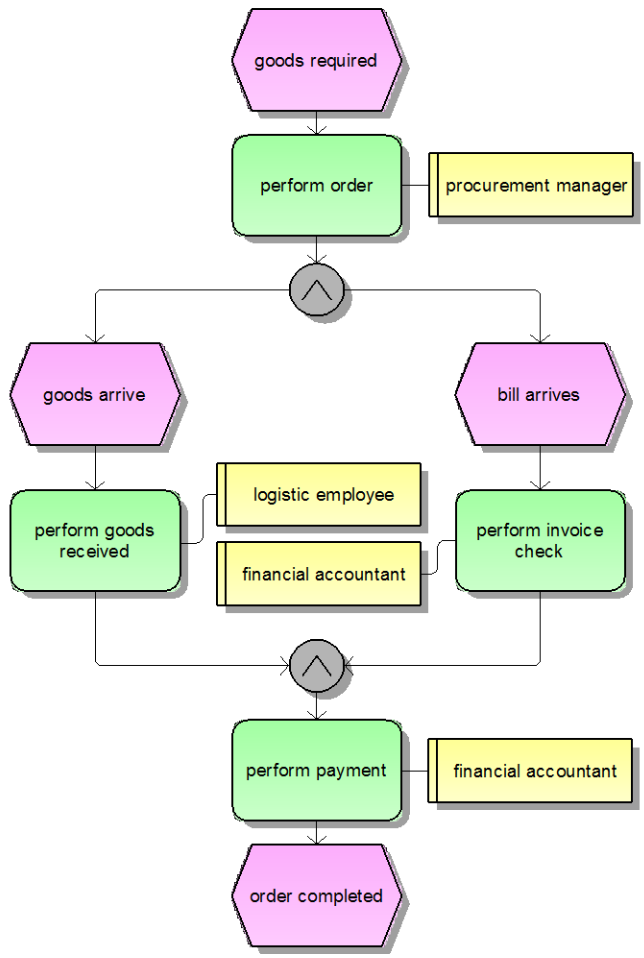


Figure 4. Procurement of Goods

C. Fragmentation of the process

The process shown in Figure 4 provides an insight into the operational sequence. Due to the simple graphical representation in EPC it is easy to understand and therefore suitable for discussions with department specialists. However, the limited amount of symbols restricts the accuracy of the representation. While the sequence with the splits and merges is precisely described by the control flow and its logical operators, details of functions and events are expressed by comments or, if supported by the tool, by additional attributes.

When we consider the modelling language EPC with respect to the layered model for knowledge transfer we can derive the following statements. The code layer contains the symbols used in the EPC diagram as well as the language in which the process is modelled. The syntactic layer contains the rules for the EPC diagram and the rules of the natural language. The semantic layer contains the connection between the words or symbols and its meaning. Because of the simple representation the precise representation depends mainly on the wording. More precise descriptions are almost impossible as the annotation of the used words is not possible. The pragmatic layer is almost not affected by the EPC. Exceptions may be additional symbols, e.g., a symbol representing the output of a process. However, the pragmatic is affected by the natural language used to describe the process and the knowledge base of the person modelling the process and the person who reads it.

The simple notation of EPC leads to a lack of precision in the semantic and pragmatic layer of the knowledge transfer. To achieve the goal of a better and ideally lossless communication in the area of business processes the descriptions concerning the semantic and pragmatic layer need to be enhanced. To achieve a better representation on the semantic and pragmatic layers the authors have decided to use frames. Every function and event in the business process will be represented as a frame.

According to Sowa [32] the frames specified by Minsky [33] are a more precise and implementable representation of the schemata. The schemata were first mentioned by Aristotle to categorize the elements of his logical arguments. Sowa [32] stated that Kant and Bartlett advanced Aristotle's schema but their definition remained too immaterial. Minsky defined a frame as a data structure to represent a consistent situation. The frame can be complemented with attributes to describe the application of the frame, the following action, or alternative actions. Minsky [33] characterizes the frame "*as a network of nodes and relations*". Minsky pointed out, that a frame has several layers and the top levels represent the true characteristics of the frame. Lower levels contain terminals that store specific data about the instance. Those instances often constitute sub- frames. With the frames Minsky intends to create an approach that imitates the human thinking in the aspect of creating pattern and apply them to new situations. He points out that a new frame often is an imperfect representation, which is gradually refined. This is facilitated by a loose coupling that enables replacement of assignments to slots.

The application of frames intends to enhance a function with a precise description. Frames allow describing a situation and changes of this situation. When used for functions the frame enables a precise description of the performance and thereby a representation of the pragmatic layer. Frames provide the opportunity to create nested structures, which allows an efficiently representation of complex situations. The inputs and outputs of functions and events, represented as frames, are described in a formal way. This aims to verify interfaces and make suggestions for modelling based on the interface verification. In addition, the semantic description should help to clarify the properties of the input and output objects. The objects describing the application of a function and the objects that represent the inputs and outputs of the function can be represented as frames too. According to Minsky they are called terminals and constitute "slots" where the data are saved. Based on the usage of the word "terminal" in computer science for an entity that cannot be further broken down, the authors will refer to the terminals of the frame as slots. Each slot can contain an object describing the characteristics of the function or an object representing an input or output of a function. Each of these objects needs to be further broken down until the costs for the break down is higher than the gained benefit. When a further break down is not possible or not reasonable anymore the object has reached the state of a terminal symbol.

D. Complement the Process

The procurement process shown in Figure 4 is composed of symbols. The symbols can be divided into function, event and logical operators. So far the features that distinguish the different types of symbols are name, shape and colour. To meet the requirements and to improve the matching of the process model with the different layers of the layered model for knowledge transfer, every function and event must be represented as a frame. The top level of such a frame contains the following slots: The frame has a name and a unique identifier (ID). The slot symbolType allows a distinction between events and functions. Both contain slots to capture all the inputs and outputs. The inputs contain everything that must be available before the corresponding object can operate. The results contain the outcome of the corresponding object. In addition to this, functions can contain slots that enclose objects describing the application of the corresponding function. Events have a slot for description instead of the one for application. Every object in the slot of a symbol can be a sub-frame itself. Thus, any level of description accuracy can be achieved.

The possibility to increase the detail depth arbitrarily is very important when applied to business process modelling. Business processes can be modelled in different levels of detail. Usually an organization decides how many levels are available for modelling and categorizing the processes.

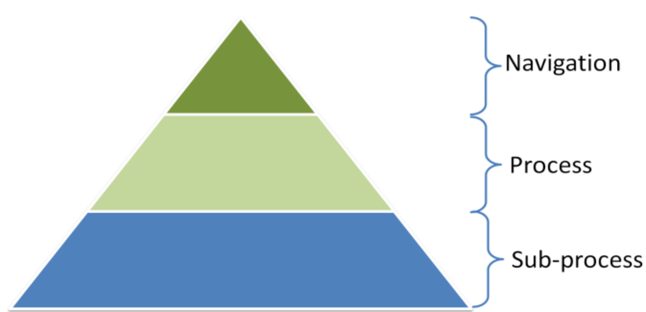


Figure 5. Process hierarchy [34]

Figure 5 displays an illustrative process hierarchy. According to Schmelzer and Sesselmann [31] the highest level constitutes a process landscape that contains the main processes with a high degree of abstraction. The upper levels in the process hierarchy aim to provide a navigation through the organizations processes. The levels below contain the business processes where sequence and execution of actions are described. A further detailing of business processes leads to the level of sub-processes. On this level the course of action is described for the sub-processes.

The need for different levels of abstraction is also highlighted by Kramer and Roovers in Scheer [34]. They argue that the need is caused by different requirements within the organization. Managers require a broad view to govern the processes and employees ask for guidance through the processes. If all these different views are implemented in a process environment the complexity requires filters to prevent an excessive demand caused by too many process models.

E. *Specification of a Function*

The navigation hierarchy is often put into practice by using value-added chain diagrams. However, for this paper the focus is on EPC and thus the lower levels of the hierarchy. The process shown in Figure 4, described as an EPC diagram, can be classified in the process layer of the process hierarchy. When describing the application of a function with inputs/outputs the next deeper level of the process hierarchy becomes visible. While normally this is accomplished by a drill down, in our model the refinement is done by representing the symbol as a frame. Figure 6 shows a frame structure for the representation of a function.

Frame name		
Slot	**Type**	**ValueType**
Slot1	<int>	ID
Slot2	<string>	name
Slot3	<symbol>	symbolType
Slot4	Dictionary<key, DB>	storages
Slot5	Dictionary<key, object>	input
Slot6	Dictionary<key, object>	application
Slot7	Dictionary<key, object>	output
Slot8	Dictionary<key, DB(object)>	outputStorage

Figure 6. Frame Structure

The frame name will contain the name of the function as shown in the EPC diagram. In order to represent a symbol in a modelling tool it is necessary to provide a unique name, in our example the ID as an integer. The name of the function is stored in another slot as string. All relevant inputs assigned to the functions are stored in a dictionary. Every object can be added to the dictionary as a pair of values, one contains a key the other the object itself. The use of keys enables a fast retrieval of objects stored in a dictionary. Moreover, dictionaries are very flexible, so it is not necessary to know the exact number of objects that the dictionary should contain. The description of the function, which describes the use and application of the function and the outputs, are also stored in a dictionary. The description objects used in the slots can be sub-frames or terminals. In Figure 6, the description objects, represented in the rows, are described by the "ValueType".

1) *Caching*

A deeper look at the representation of inputs and outputs in functions and events highlights another issue. When the inputs and outputs of every function and event are described, the amount of necessary inputs and outputs increases rapidly. This is caused by passing on the input / output objects, even if they have no influence on the current process step. This passing on might be necessary if an process step on the end of the process requires an input that is not used by the preceding process steps but which is available from the beginning. Referred to our example process this might be account information, which are recorded in another process but used to perform the payment. This passing on of inputs and outputs has a negative effect on the reusability of the functions.

To solve this problem the authors suggest using storage, as it is also used in the eEPC diagram, to cache the object and reuse it in another process step. Such storage could be a data base system, in a less automated landscape a manual storage such as an address book or a conceptual storage.

2) *Application to the Event Perform Order*

In the following the business process "Procurement of Goods" will be enhanced by using frames to refine the description of functions and events. The process starts with the event "goods required" shown in Figure 7. This event contains a human readable description and the output: "RequirementQuantityGoods". This output is a sub-frame that contains a 'requirement' for a 'quantity' of 'goods'. The output of the event "goods required" is also the input for the first function "perform order". Figure 8 shows the function "perform order" represented as a frame.

Goods Required		
Slot	**ValueType**	**Value**
Slot1	ID	1
Slot2	name	"goods required"
Slot3	symbolType	event
Slot4	input	
Slot5	Description	"This event occurs when the demand for goods is greater than the available quantity"
Slot6	output	1, RequirementQuantityGoods

Figure 7. Frame: Goods Required

Perform Order		
Slot	**ValueType**	**Value**
Slot1	ID	2
Slot2	name	"perform order"
Slot3	symbolType	function
Slot4	storages	1, SAP_6
Slot5	input	1, RequirementQuantityGoods; 2, SupplierPriceGoods
Slot6	application	1, CreateOrder
Slot7	output	1, OrderNumber
Slot8	outputStorage	1, SAP_6(OrderNumber)

Figure 8. Frame: Perform Order

As shown by the figure above, the function has another input, which is required for the application. However, the input "SupplierPriceGoods" is not an output of the previous event. This second input is a sub-frame that contains a 'supplier' and a 'price' for the 'goods'. If this input cannot be delivered by the previous function there must be another way to supply it to the function. At this point, the storage is used.

Data that are used by many processes of a company are commonly stored in a central storage. For our example process this storage is an enterprise resource planning (ERP) system, the SAP ERP system (SAP_6). Enabled by the connection to this storage, mapped in slot 4 of the frame, the function can search for the required data and use them if available. To make this perfectly clear, the connection described here is not a connection to the real SAP server in the company. It is, however, a representation of the real storage and the data that are stored there.

Order		
Slot	**ValueType**	**Value**
Slot1	ID	201
Slot2	order number	000213736
Slot3	description	"An order is a legally binding request for goods to a supplier and contains a unique order number. "
Slot4	necessary elements	1, goods; 2, supplier; 3, customer; 4, price; 5, delivery date; 6, date of order
Slot5	Transaction code	"ME21N"

Figure 9. Frame Order

The data in the storage are used as input for a function or are created as output of a function. The description of the application is contained in slot 6. The function "perform order" contains the sub-frame "CreateOrder". Depending on the purpose the process model is used for, the frame could contain further descriptions for persons or computers. Descriptions for persons may contain instruction manuals, transaction codes to perform the action at the system or a list with all required inputs and created outputs. For machine processing it may contain code that can be execution directly by a computer. Output of the function is the sub- frame "Order", which is output for the next process step but also stored in the storage SAP_6. Figure 9 shows the "Order" frame with its slots. The properties stored in the slots are chosen with the target to create understanding of the object and to facilitate a later implementation in software.

F. *Modelling Support, Analysis, and Optimisation*

The application of the layered model for knowledge transfer aims to support both, modelling and usage of the business process model. The modelling should benefit by automatic syntax checks, verifying the model against the modelling rules. However, such syntax checks are already implemented in various modelling environments. Furthermore, the modelling environment should generate recommendations for the subsequent process step if an appropriate element exists in the database. An important point for this suggestion is constituted by the descriptions of the outputs of the current process step. The system will analyse the process and search for a suitable subsequent process step with matching inputs. Accordingly the system will suggest the object found during the search to the modeller where the conformity of the interfaces is indicated in percent. Thus, the modeller can make the decision whether to use the suggested element, which might be a function or an event.

It has to be considered that process models pursue different targets. For process models used as work instruction the semantic annotation and enhanced descriptions can constitute a benefit. So, for example, the components of an order, used in the "process procurement of goods", can be looked up as illustrated by Figure 9.

For the optimisation of a process the description of inputs and outputs as well as the description of the application are of great importance. Based on this various optimisation approaches could be undertaken.

VI. Conclusion and Future Directions

Knowledge transfer is affected by many different parameters. Due to of the relevance of knowledge transfer, it is important to understand the impact of the different parameters. The sociologists Luhmann and Aufermann deal with communication aspects but they neglect the issue of a practical implementation. Shannon's model focuses on the technical implementation but is restricted to the layers of code and syntax. The model of Nonaka and Takeuchi deals with organizational knowledge and knowledge conversion, but the practical transmission is not considered in detail. Ammann describes knowledge conversions in more detail.

However, this model is still too abstract to facilitate its implementation. The approach presented in this paper addresses these issues by introducing a model with different layers.

The application of the model focuses on the transfer of knowledge by using written language as a medium. While the description of knowledge transfer is not very precise in the approaches of Nonaka and Takeuchi our approach goes more into detail. The further refinement of the knowledge transfer, during the externalisation and internalisation intends to create a profound understanding about the process of knowledge transfer. Another intention behind introducing the layers is to reduce errors on each of the knowledge levels. Thus, the process of knowledge transfer is divided into several steps, which can be examined separately. This makes it easier to detect and identify errors and facilitates the prevention of misinterpretation.

The application of the layered model for knowledge transfer in the area of business process management shows one possible area of application. The different knowledge base of the person modelling the process and the employees in a department running the process constitutes a challenge for both, modeller and employee. Due to the more detailed description of the process by using frames, which integrate the semantic and the pragmatic layer, this challenge is addressed.

In this paper, the authors presented a layered model for knowledge transfer and applied it to the area of business process modelling. However, at this stage the application of the model is a theoretical model; the application is still to be proven. Therefore, the next step will be to develop a prototype application for modelling of business processes and run a validation. Goal of the investigation will be to determine the effect on the modelling and the analysis of business processes through the use of the layered model for knowledge transfer.

REFERENCES

[1] F. Schiele, F. Laux, and T. M. Connolly, “A layered model for knowledge transfer,” in SEMAPRO 2013, The Seventh International Conference on Advances in Semantic Processing, 2013, pp. 26–31.

[2] K. Dalkir, *Knowledge management in theory and practice*: MIT Press, 2011

[3] T. H. Powell and V. Ambrosini, “A pluralistic approach to knowledge management practices: Evidence from consultancy companies,” Long Range Planning, vol. 45, no. 2–3, 2012, pp. 209–226, http://www.sciencedirect.com/science/article/pii/S002463011200009X, [accessed June 2013].

[4] I. Nonaka and H. Takeuchi, *The knowledge-creating company*. Oxford: Oxford University Press, 1995.

[5] I. Nonaka, “A dynamic theory of organizational knowledge creation,” Organization Science, vol. 5, 1994, pp. 14–37, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.115.2590, [accessed May 2014].

[6] J. Rowley, “The wisdom hierarchy: representations of the DIKW hierarchy,” J. Information Science, vol. 33, no. 2, 2007, pp. 163–180.

[7] U. Hasler Roumois, *Studienbuch Wissensmanagement: Grundlagen der Wissensarbeit in Wirtschafts-, Non-Profit- und Public-Organisationen*, translated: *Record of study knowledge management: Foundations of knowledge work in business-, non-profit- and public-organization*, 2nd ed. Zürich: Orell Füssli, 2010.

[8] T. H. Davenport and L. Prusak, *Working knowledge: How organizations manage what they know*. Boston, Mass: Harvard Business School Press, 2000.

[9] M. Boisot and A. Canals, “Data, information and knowledge: have we got it right?,” Journal of Evolutionary Economics, vol. 14, no. 1, 2004, pp. 43–67.

[10] R. K. Rainer and C. G. Cegielski, *Introduction to information systems*, 3rd ed. Hoboken, N.J, Chichester: Wiley; John Wiley [distributor], 2011.

[11] R. L. Ackoff, “From data to wisdom,” Journal of Applied System Analysis, vol. 16, 1989, pp. 3–9.

[12] M. Frické, “The knowledge pyramid: a critique of the DIKW hierarchy,” J. Information Science, vol. 35, no. 2, 2009, pp. 131–142.

[13] M. Weggeman, *Wissensmanagement: Der richtige Umgang mit der wichtigsten Ressource des Unternehmens*, translated: *Knowledge management: The right way to deal with the most important resource of the company*, 1st ed. Bonn: MITP-Verl, 1999.

[14] J. S. Valacich, C. Schneider, and L. M. Jessup, *Information systems today: Managing in the digital world*, 4th ed. Upper Saddle River, N.J: Prentice Hall, 2010.

[15] G. B. Davis and M. H. Olson, *Management information systems: Conceptual foundations, structure, and development*, 2nd ed. New York: McGraw-Hill, 1985.

[16] F. I. Dretske, *Knowledge and the flow of information*. Stanford, CA: CSLI Publications, 1999.

[17] M. Polanyi, *The tacit dimension*. London, England: Cox & Wyman Ltd, 1966.

[18] I. Nonaka, P. Byosiere, C. C. Borucki, and N. Konno, “Organizational knowledge creation theory: A first comprehensive test,” Organization Science, 1994.

[19] I. Nonaka, “The knowledge-creating company,” Harvard Business Review, vol. 69, no. 6, 1991, pp. 96–104.

[20] E. Ammann, “A hierarchical modelling approach to intellectual capital development,” in Electronic Journal of Knowledge Management, vol. 8, Issue 2, C. Bratianu, Ed, 2010, pp. 181–191.

[21] M. Spitzer, *Lernen: Gehirnforschung und die Schule des Lebens*, translated: *Learning: Brain Research and the School of Life*, 1st ed. München: Spektrum Akademischer Verlag, 2007.

[22] H. Krcmar, *Informationsmanagement*, translated: *Information management*, 5th ed. Berlin, Heidelberg: Springer, 2010.

[23] N. Luhmann, *Soziale Systeme: Grundriss einer allgemeinen Theorie*, translated: *Social Systems: Outline of a general theory*, 1st ed. Frankfurt am Main: Suhrkamp, 1987.

[24] M. Berghaus, *Luhmann leicht gemacht: Eine Einführung in die Systemtheorie*, translated: *Luhmann made easy: An Introduction to Systems Theory*, 3rd ed. Köln: Böhlau, 2011.

[25] J. Aufermann, *Kommunikation und Modernisierung: Meinungsführer und Gemeinschaftsempfang im Kommunikationsprozess*, translated: *Communication and modernisation: opinion leaders and community reception in the communication process*, München-Pullach: Verlag Dokumentation, 1971.

[26] C. E. Shannon, “A mathematical theory of communication,” SIGMOBILE Mob. Comput. Commun. Rev, vol. 5, no. 1, 2001, p. 3-55.

[27] A. Roch, *Claude E. Shannon: Spielzeug, Leben und die geheime Geschichte seiner Theorie der Information*, translated: *Claude E. Shannon: Toys, life and the secret history of his theory of information*, 1st ed. Berlin: Gegenstalt, 2009.

[28] H. Zimmermann, "OSI reference model--The ISO model of architecture for open systems interconnection," IEEE Transactions on Communications, vol. 28, no. 4, 1980, pp. 425–432, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1094702, [accessed May 2014].

[29] A.-W. Scheer, *ARIS--business process modeling*, 3rd ed. Berlin, New York: Springer, 2000.

[30] G. Keller, M. Nüttgens, and A. W. Scheer, "Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK)," translated: "Semantic process modelling based on event-driven process chains (EPC)", Universität des Saarlandes, Germany, Saarbrücken, 1992.

[31] H. J. Schmelzer and W. Sesselmann, *Geschäftsprozessmanagement in der Praxis: Kunden zufrieden stellen – Produktivität steigern – Werte erhöhen*, translated: *business process management in practice: satisfy customer – increase productivity – increase values*, 8th ed. München: Hanser, 2013.

[32] J. F. Sowa, *Knowledge representation: Logical, philosophical, and computational foundations*. Pacific Grove, Calif. [u.a.]: Brooks/Cole, 2000.

[33] M. L. Minsky, *A framework for representing knowledge*. Cambridge: Massachusetts Institute of Technology A.I. Laboratory, 1974.

[34] A.-W. Scheer, Ed, *Agility by ARIS business process management*: With 2 tables. Berlin, Heidelberg, New York: Springer, 2006.

Detection and Classification of Anomalous Events in Water Quality Datasets Within a Smart City-Smart Bay Project

Dian Zhang*†, Timothy Sullivan†, Ciprian Briciu-Burghina†, Kevin Murphy†, Kevin McGuinness*, Noel E. O'Connor*, Alan Smeaton*, and Fiona Regan†

*CLARITY: Centre for Sensor Web Technologies, Dublin City University, Glasnevin, Dublin 9

† MESTECH: Marine and Environmental Sensing Technology Hub, Dublin City University, Glasnevin, Dublin 9

Email: dian.zhang2@mail.dcu.ie, tim.sullivan@dcu.ie

ciprian.briciu@dcu.ie, kevin.murphy@dcu.ie, kevin.mcguinness@dcu.ie

noel.oconnor@dcu.ie,alan.smeaton@dcu.ie,fiona.regan@dcu.ie

***Abstract*—Continual measurement is key to understanding sudden and gradual changes in chemical and biological quality of water, and for taking reactive remedial action in the case of contamination. Monitoring of water bodies will increase in future within Europe to comply with legislative requirements such as the Water Framework Directive and globally owing to pressure from climate change. Establishing high quality long-term monitoring programs is regarded as essential if the implementation of pertinent legislation is to be successful. However, conventional discrete sampling programs and laboratory-based analysis techniques can be costly, and are unlikely to provide timely and reliable estimates of true ranges of deterministic physicochemical variability in a water body with marked temporal or spatial variability. Only continual or near continual measurements have the capacity to detect ephemeral or sporadic events, thus providing the potential for on-line event detection and classification. The aim of this work is to demonstrate the potential role of continuous data acquisition in decision support as part of a smart city project. In this work, a multi-modal smart sensor network system framework for large scale estuarine and marine water quality monitoring is proposed. The application of a number of evolving techniques that allow automated detection and clustering of events from data generated by *in situ* sensors within environmental time series datasets is described. We provide examples of how change in the range of variables such as turbidity and salinity might be detected and clustered to provide the end user with greater ability to detect the onset of environmentally significant events. Finally, we discuss the acquisition of data from *in situ* water quality sensors and its utility within the framework a smart city-smart bay integrated project.**

Keywords–Continuous water monitoring estuary, marine, decision support, turbidity, salinity, anomaly detection, robust online clustering, pixel-based adaptive segmentation.

I. INTRODUCTION

Automated collection and storage of datasets related to environmental water quality is now becoming commonplace, however, challenges remain in automated detection of important events within these datasets and thus determination of the value and ecological significance of collected data for use in decision support systems [1]. This challenge can only increase as the vision of futuristic smart cities containing integrated sensing networks becomes a reality. *In situ* sensors capable of continually sampling chemical and physical parameters offer the potential to reduce costs, provide timely information and improved representation of long-term trends in the fluctuations of pollutant concentrations [2]. *In situ* in the context of environmental sensing means in place or in direct contact with the medium of interest, as opposed to methods such as remote sensing where no contact is made between the sensor and the analyte. Indeed, the ideal aquatic monitoring system of the near future might consist of a network of sensors deployed at key locations, capable of autonomous operation in the field for a year or more [3][4]. Despite the increasing range of techniques available, continuous on-line *in situ*, measurement systems remain largely limited by both poor sensor performance and a lag in availability and application of suitable data analytics. Thus, while measurement and detection methods exist for many environmental pollutants in the laboratory, continuous monitoring on a cost effective basis in the field remains a challenge for these reasons.

A. *The Ideal System*

The ideal monitoring system of the near future might consist of sensor networks deployed at key locations, capable of near autonomous operation in the field over long time frames (annual to decade time scales). The components necessary to achieve this measurement of multiple parameters, simultaneously and in real-time are available [5]. However, it is clear that as a scientific community, we need to improve the quality and reduce the cost of sensors for many of the desired parameters (for example most nutrients, microbial contaminants), while using simplified devices in robust embedded networks to make this ideal truly achievable. Another consideration is that a common platform for data validation and sensor verification has yet to be universally implemented to improve data quality. Data collected from monitoring stations can be communicated by wireless technology prior to statistical processing and interpretation by expert systems. Indeed wireless data transmission, and the concept of wirelessly networked sensors in particular, has however become one of the most dynamic and important areas of multi-disciplinary research [6][7]. Real time alerts can be raised to relevant personnel, perhaps through an alarm sent to smartphones or e-mail, when trends for any constituent of interest breaches particular thresholds (for example Environmental Quality Standards (EQS)) are detected. Notified personnel can then intercept serious pollution incidents or lead an appropriate response. Detected individual outliers can be combined into event-based information to support the identification of impacts from environmental threats. Events can be further clustered into groups based on some kind of similarity matrix to assist scientists in identifying commonality between groups.

B. The Future: improved signal processing including both on-line and off-line data analytics

Implementation of advanced and user-friendly event-detection software to distinguish between normal conditions and anomalous events is critical if data provided by the ideal sensing system is to be used effectively. Data-driven estimation models with sequential probability updating have been suggested for this purpose [8] and implemented in various forms (see for example CANARY [9][10][11]). Detection of water security threats arising either from intentional or unintentional sudden contamination events implemented in Contamination Warning Systems (CWS) is of particular importance, and it has been suggested that in the region of thirty-three contaminants (pesticides, insecticides, metals, bacteria, etc.) can be utilised as indicators of intentional water contamination [12]. While widespread detection of all thirty-three variables using low-cost autonomous sensor systems is far from achievable at present, the potential advantages of automated early warning systems based on multivariate analysis of datasets collected by such systems are clear. Moreover, considerable expenditure of research effort is required to further develop improved data analytics platforms for the ideal sensing system, including forecasting, modelling and event detection platforms or Early Warning Systems (EWS) [13]. The rapid growth of “big data” provided by social media, concomitant with improved computing capacity has spurred research interest into novel data analytics techniques. Datasets provided by *in situ* sensing in its current form are at present not approaching the scale of those provided by social networking scenarios, however, as the vision of internet-scale sensing heralded by the development of improved sensing becomes reality, such datasets may become widespread. Detection of anomalous events within these datasets would be of widespread interest and a number of Artificial Intelligence (AI) methods, such as artificial neural networks (ANN) and support vector machines (SVM), have already been utilised for this purpose (see for example [14][15][16] and [17] for a critical review of ANN usage in this regard). Generally, these techniques have been used to classify water quality data into normal and anomalous classes after supervised learning training. Other data-mining methods, such as K-means classification and the multivariate nearest-neighbour (MV-NN) algorithms, combining different water-quality parameters and location information, are also used for protecting drinking water systems [18]. Data-fusion methods have been used to combine various types of information, for example, operational data or data from multiple monitoring stations [19] or sensors [20] to improve the detection of water-contamination events while reducing the potential number of false positives. Other approaches have proposed combining residuals for water-quality parameters with autoregressive (AR) models or other methods (see for example [20][21]). An extension of this is to use some form of pattern recognition and matching to detect and and create a multivariate library of known events. A newly detected event can then be matched to the library of historical events to determine if similar events have occurred previously. If so, information gained from historical events can be used to analyse the causes and impacts of current ongoing events. Event clustering would be a typical approach to this problem and has been implemented in several such systems (for example CANARY uses a trajectory clustering-based pattern matching approach). However, existing systems such as CANARY, are focusing on contamination event detection for drinking water systems which are very different from marine or estuarine environments. Drinking water systems normally have a closed supply chain and dare not affected by many factors such as weather, tide, season, dam release, etc., in contrast to open water bodies. A key challenge for on-line automated analysis of environmental datasets lies in dealing with the peculiarities of these datasets themselves, in which missing values are common, disjointed measurement methodologies and techniques are followed or in which large-scale uncertainty can exist due to sensor performance issues. Successful development of useful early warning systems and other on-line data analytics methods for *in situ* sensing platforms must be capable of dealing with these issues. Particular issues of current early warning systems include high proportion of false alarms and false negatives in practical applications, unacceptable computational demands and lack of on-line detection [22].

This work outlines the potential for continuous water quality monitoring in decision support as part of a Smart Bay component [23][24] in the broader context of a connected Smart City project in Dublin. Over the coming years, the SmartBay project will see the expansion of a multi-modal sensor and data network in Dublin Bay for detection of pollution and flood events among others. The latter will consist of a number of sensor deployments, including visual sensing systems, modelling and integration of additional available data sources (for example data provided by citizen monitoring). Datasets collected over the course of the SmartBay project can be utilised for other applications depending on user requirements or emerging applications, with particular emphasis on water in the city, port and coastal areas. In this paper, real data collected from pilot sites in Dublin Bay using continuous autonomous multi-parameter sensing systems are used to demonstrate how machine-learning techniques such as robust online clustering (ROC) and a modified pixel-based adaptive segmentation (MoPBAS) approach can be utilised for such purposes. These techniques will be discussed in terms of anomaly detection, event construction and classification, and the resulting opportunities for development of decision support systems. We show how use of a multi-modal data system demonstrates potential for low-cost sensing in complex aquatic environments such as estuaries. Data from water quality sensors are evaluated and analysed along with data from grab samples, with the latter supporting the observations of trends from water quality monitoring systems. Scenarios presented provide examples of the potential value of such a monitoring system in building a SmartBay infrastructure. The rest of the paper is organised as follows. Section II presents the proposed multi-modal smart sensor network framework for marine environmental monitoring. Section III introduces the pilot site and the deployment of instruments. The implementation of anomaly detection, abnormal event construction and clustering is described in Section IV and the experiment results are shown in Section V. The conclusion of the paper is in Section VI.

II. Multi-Modal Abnormal Event Detection and Classification Framework for Smart-Bay Monitoring System

In order to fully utilise the above ideal system, a novel multi-modal smart sensor network framework for marine environmental monitoring is proposed. Figure 1 illustrates an overview structure of the system. The framework consists of three layers, a wireless sensor network (WSN) layer, backend smart system layer and a decision-making layer. The WSN layer contains observation sites that are equipped with various numbers and type of sensors. The smart system layer has two main components. The data repository collects data generated from all sensors and creates a truly multi modality data pool. The smart system processes these data sets using state-of-the-art machine learning techniques to convert raw sensor measurements into organised knowledge that can be understood by operators. The operators can then make decisions based on this information to avoid or reduce negative impacts. Moreover, the operators can then send feedback to the WSN to indicate whether the current deployed sensor network is sufficient to monitor and subsequently model the observation sites. The overall architecture of the deployments made around Dublin bay is shown schematically in Figure 1.

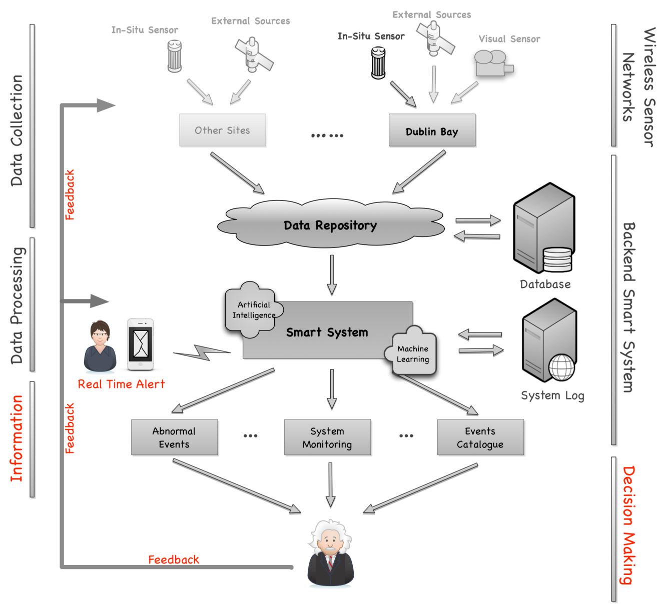


Figure 1. A schematic outlining the architecture of the proposed multi-modal smart monitoring system, including treatment of the data and the feedback mechanisms for decision-making.

This architecture is flexible and extendable allowing other sites and data sources to be added without overly increasing complexity. The data from *in situ* sensors at Dublin Bay can be complemented with data from external sources, for example weather forecasts, whereas other sensing modalities (such as visual sensing) can also provide information on human activities, allowing distinction between anthropogenic and natural re-suspension events. The system can also send real time alerts to operators, so that they can react quickly to avoid or limit negative impacts. In terms of data analysis, the system can be formed as data collection, data process and information stages. The data collection process involves deploying and maintaining the wireless sensor networks, where the data processing level converts raw data into information easily interoperated by operators. The information stage maintains a large indexed content based archive, which allows the user to browse and query events.

III. Monitoring Location and Pilot System Deployed

The following describes Dublin Bay, the site used as the location for this study, along with the equipment used for collection of continuous monitoring data at the site.

A. Test Site

Dublin Bay (latitude: 53°20’39”, longitude: -6°12’59”) is located on the lower Liffey Estuary Dublin Ireland in a busy port environment (see Figure 2). The estuary is a diverse ecosystem with many micro-environments that include benthic communities, fish and shellfish, seabird populations and marine mammals [25][26]. The area is also a zone of passage for salmon and sea trout migrating to and from feeding and spawning areas [27]. The topography of the estuary has been greatly modified, and is constrained by walls along its whole length and is regularly dredged to remove accumulated sediments. The working site is located in the upper part of the Estuary, where the ship traffic is less intensive. Average water depth in the area is approximately $8m$ and the width of the channel is approximately $260m$. Due to the large amount of activity at the site and its importance from an environmental and ecological perspective, the site was equipped with a multi-parameter *in situ* sensor along with a visual sensing system.

Figure 2. Overview of the Dublin Bay area, indicating the location of the deployed pilot system , which provided the datasets used in this work. Dublin Bay image source: Google Maps. Retrieved: 2014-04-11

B. Instrumentation

A multi-parameter sonde (YSI 6600EDS V2-2), equipped to measure turbidity (Nephelometric Turbidity Units(NTU)), optical dissolved oxygen (mgL^{-1}/%saturation), temperature (°C), conductivity ($mScm^{-1}$), depth (m) and telemetry system (EcoNet) was purchased from YSI Hydrodata UK. The sonde was deployed at a depth of $2.5m$ from the water surface, and data was collected since 1st of Oct 2010 with a sampling interval of $15mins$. Temperature, dissolved oxygen and salinity were checked using a ProPlus handheld multi-parameter instrument (YSI Hydrodata UK) and turbidity was validated using a portable turbidity meter Turb® 430 IR (VWR Ireland). Both hand held instruments were calibrated in the laboratory prior to site visits as per manufacturer's protocols. Site visits were undertaken fortnightly in winter and weekly in spring. Copper tape and mechanical wipers (for the optical oxygen and turbidity sensors) were used to control biofouling of sensor systems.

IV. Methodology

To detect and cluster environmental events, anomalous sensor readings (also referred as outliers) need to be extracted from a continuous data stream. These abnormal sensor measurements are then grouped into events based on proximity in time. A set of features is extracted that is characteristic of different anomalies and is used to assign individual events. Each event might have different temporal characteristics; so to compare their similarities, a bag-of-words approach is adopted to encode these features as constant length descriptors. Each feature set of the detected anomalies is matched against a predefined codebook and the closest matching codeword is used to represent the feature. The event is then represented by the frequency of occurrence of each word. Once the feature vector of the event is constructed, a clustering method is applied to group these events into subclasses based on their similarities. Figure 3 shows the flow diagram of the proposed framework. Each step of the proposed framework is introduced in detail as follows.

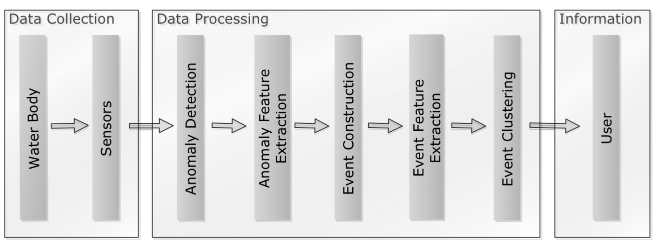


Figure 3. Flow diagram of the proposed system framework

A. Anomaly Detection

To detect abnormal events, we first need to detect unusual sensor measurements in the data stream. An unusual or anomalous sensor measurement, is defined as a sensor reading that differs considerably from recent observations. Thus, an anomaly can be detected by modelling previous sensor measurement trends. To achieve this, we have modified the pixel-based adaptive segmenter (MoPBAS) method originally proposed by Martin Hofmann et al. for image segmentation [28]. A non parametric water quality background trend model is built based on a history of recently observed sensor readings. The classification of an unusual reading depends on a decision threshold, which is adapted based on the variations in the data stream. The model is updated over time according to the dynamics of the measurements. In the following, we describe the process by which the MoPBAS method is used to detect abnormal sensor readings.

1) Background Trend Model and Anomaly Classification: To classify a new incoming value $I(t)$, a sensor reading trend model $B(t)$ is built. $B(t)$ is defined by an array of N recently observed values.

$$B(t) = \{B_1(t), ..., B_k(t), B_N(t)\} \tag{1}$$

In [28], incoming values are classified based on the total number of distances between input value $I(t)$ and all elements in $B(t)$ that are smaller than threshold $T(t)$. We found that just comparing the minimum distance with the threshold is sufficient to differentiate the measurements.

$$I(t) = \begin{cases} 1, & \text{if min(dist}(I(t), B_k(t))\text{)} > T(t) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

If the input value is classified as normal ($I(t) = 0$), it can be used for updating the background trend model. The update probability depends on the learning rate $L(t)$.

2) Update of the Decision Threshold: When monitoring water quality of estuarine waters, there can be periods of time where large variations occur in measured variables, such as after heavy rainfall, and time periods with little change or fluctuation. Ideally, for periods of high variability, the threshold $T(t)$ should be increased and for stable conditions, $T(t)$ should be decreased. To quantify this dynamic, the mean $\overline{d}_{min}(t)$ of the previous N minimum distances between input values and trend model are calculated as the measure of the trend variations. For instance, assuming the water quality measurements remain constant, $\overline{d}_{min}(t)$ will be zero. In contrast, $\overline{d}_{min}(t)$ will be higher for more dynamic backgrounds. The decision threshold can then be adapted as follows:

$$T(t) = \begin{cases} T(t) \times (1 - T_{inc/dec}), & \text{if } T(t) > \overline{d}_{min}(t) \times T_{scale} \\ T(t) \times (1 + T_{inc/dec}), & \text{otherwise} \end{cases} \tag{3}$$

where $T_{inc/dec}$ is a static value that controls the threshold update rate and T_{scale} is also a fixed parameter, which stretches $\overline{d}_{min}(t)$ to the same range as $T(t)$. T_{lower} and T_{upper}, which are also fixed values, control the upper and lower bounds of the threshold, thus the threshold will not grow out of range.

3) Update of the Learning Rate: Another important parameter of MoPBAS is the trend model learning rate L_t. Water quality measurements have characteristics that are significantly different from image segmentation data. Values measured by *in situ* sensors are typically very noisy, have lower sampling rates (in terms of minutes compares to fraction of a second in the image processing domain) and vary from a baseline (they change gradually due to "global" effects, such as wind, tide etc.). Unlike background modelling in the image processing domain, in which foreground objects will be slowly merged into the background if it no longer moves, water quality parameters will usually return to a baseline level after an event. Thus, we normalise ($R(t)/R_{upper}$) and invert the learning rate proposed in the original PBAS method. Here, the learning rate is defined as follows:

$$R(t) = \begin{cases} R(t) + \frac{L_{inc}}{\overline{d}_{min}(t))}, & \text{if anomaly = true} \\ R(t) - \frac{L_{dec}}{\overline{d}_{min}(t))}, & \text{if anomaly = false} \end{cases} \tag{4}$$

$$L(t) = 1 - R(t)/R_{upper} \tag{5}$$

Where L_{inc} and L_{dec} are fixed values that control the increasing and decreasing intervals. The variation in $R(t)$ is limited by an upper and lower bound: $R_{lower} < R(t) < R_{upper}$. The learning rate also depends on the background dynamics ($\overline{d}_{min}(t)$). When an event occurs, measured values provided by the sensor will usually deviate greatly from the baseline level. Thus, the trend model should be updated slowly or not updated at all. In contrast, after an event occurs, sensor readings will usually stabilise or return to the baseline, and the trend model should be updated quickly. When an anomaly is first detected ($\overline{d}_{min}(t)$ is small), $R(t)$ increases rapidly, thus the learning rate $L(t)$ decreases sharply. However, $\overline{d}_{min}(t)$ will become large quickly when multiple anomalous readings are detected, which results in $R(t)$ and indeed $L(t)$ remaining constant or only changing slightly. When sensor readings stabilise or return to a normal range, $\overline{d}_{min}(t)$ becomes small and $L(t)$ will increase.

4) Update of the Trend Model: Updating the trend model, B, is essential to capture global effects, such as tide or wind. The learning rate $L(t)$ is used as the update probability and an element in the trend model is randomly chosen and replaced by the incoming value. However, this process is only performed when no anomalous values are detected. This allows the incoming sensor measurement to be "learned" and incorporated into the trend model. In the original PBAS, a randomly chosen neighbouring pixel is also updated, however, as there is no "neighbour" (image data is 2D as opposed to 1D water quality data) and this step is not performed.

5) Distance Calculation: Rather than using common distance metrics, such as Euclidean distance, we use the root of the absolute square difference (RASD) to calculate the distance between incoming value and the ith element in the trend model.

$$D_i(t) = \sqrt{|I(t)^2 - B_i(t)^2|} \tag{6}$$

Figure 4 shows the ratio between our distance metric and the 1-D Euclidean distance (for illustration purposes, the input $I(t)$ range is set from 5 to 104 in steps of 1, background $B_i(t)$ is set to 5). It can be seen from the graph that when the distance is large, the output is approximately equal to the 1-D Euclidean distance. However, the output is enhanced when the different between $I(t)$ and $B_i(t)$ is small. This is a key factor when calculating the background dynamic $\overline{d}_{min}(t)$, as it smooths the effect of an event to $\overline{d}_{min}(t)$. Thus, the value of $\overline{d}_{min}(t)$ will not increase rapidly when an event occurs as shown.

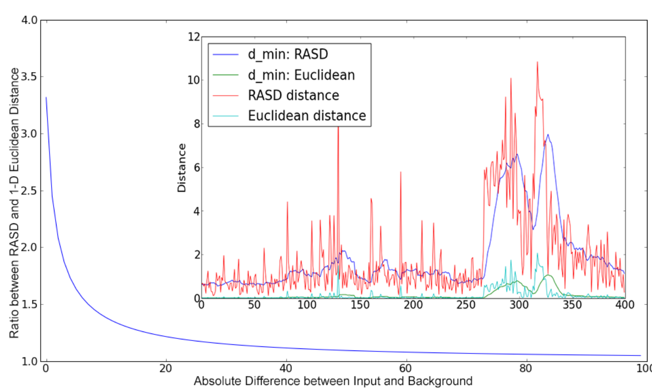


Figure 4. Demonstration of the ratio between RASD distance metrics and 1-D Euclidean distance, the inner graph shows RASD distance method, which enhances small distances, smoothing the variation of the background dynamics $\overline{d}_{min}(t)$.

B. Anomalous Feature Extraction

To capture the similarity in anomalies detected, and for further clustering of anomalous events, we need to extract a set of features that are sufficiently discriminative to allow us to classify unusual readings and subsequent events. The feature set of an anomalous reading has the following components: the difference between the previous sensor measurement $I(t-1)$ and current sensor measurement $I(t)$, current sensor measurement $I(t)$, the difference between current sensor measurement $I(t)$ and the next sensor measurement $I(t+1)$, the minimum distance between sensor measurement and trend model d_{min}, and the distance between the minimum distance d_{min} and the threshold T(t). The feature set $f(anomaly)$ can be represented as:

$$f = [I(t-1) - I(t), I(t), I(t) - I(t+1), d_{min}, d_{min} - T(t)]$$

C. Event Constructing

Anomalies detected by the MoPBAS method are grouped into events according to their temporal information. To achieve this, agglomerative hierarchical clustering is applied. As shown in Figure 5, consecutive anomalies are combined together into a single event. Furthermore, if the gap between a new anomaly and previous outlier is smaller than a threshold, T_{gap}, the new anomalous value will be merged into the same event. In contrast, if this gap is greater than T_{gap}, a new event will be created.

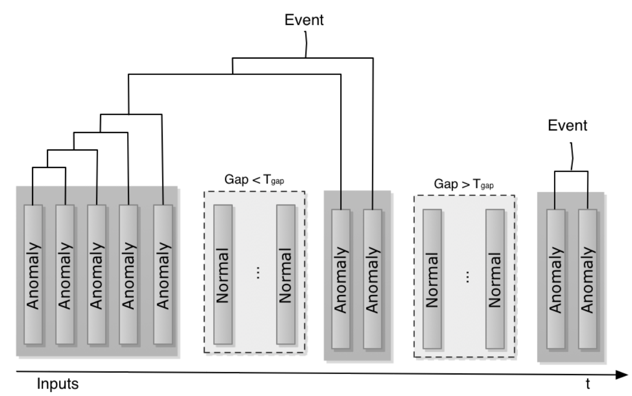


Figure 5. Anomalies are grouped into events using agglomerative hierarchical clustering based on their temporal information

D. Event Clustering

A Bag-of-Words approach is widely used in text document classification [29], content-based image retrieval [30] and image recognition tasks [31], where a document is represented as a bag of its "words" or a bag of small image patches (visual words) in the image processing domain. Most classification or clustering methods require a fixed number of feature dimensions. However, for many tasks, such as text document indexing, the number of features extracted from each file are generally different. The Bag-of-Words method represents these features by counting the frequency of occurrence of each "word" as the descriptor of the object. For text document processing, a "word" generally means an entry in a "codebook", which is the combination of a single word in a dictionary or a phrase. In the image processing domain, a word (some times referred as a "visual word") means a small image patch or fragment. As each environmental event may contain a different number of anomalous values, each outlier feature set is represented by a "sensor word" in order to quantify the similarities between events, and the frequency of their occurrence is reconstructed as the descriptor of the event. To create a codebook, K-means clustering is performed over a set of training data. The centres of the learned clusters are then defined as codewords. Each anomaly feature set in an event is mapped to a certain codeword in the codebook and the event can be represented by the histogram of the occurrence of the codewords.

To divide events into groups, a clustering method known as robust on-line clustering [32] is used. Clustering is the process of dividing instances into groups in such a way that instances in the same group are more similar than elements in other groups. There are many common clustering methods that are widely used such as K-Means or Mean-shift. Current research indicates that there is no known single clustering method that categorically out performs all others in all tasks. The benefit of using robust on-line clustering in this context is that, unlike K-Mean or Mean-Shift, this method is not sensitive to "noisy" data. This is a key requirement for environmental monitoring tasks where highly variable data could indicate a significant event. Moreover, robust on-line clustering is an on-line method that can be used to process a continuous data stream provided by *in situ* sensors.

V. Experiments and Results

This section describes the experiments that carried out to evaluate the proposed system. The initial value of all parameters in the proposed approach are also listed in this section. As a proof of concept, we are focusing on salinity and turbidity measurements, however, the same framework may be applicable to other water quality parameters that have similar time series characteristics.

A. Test Data

The dataset that is used for evaluating the proposed system was collected from deployed remote water quality monitoring systems in Dublin Bay between 1st Oct 2010 and 3rd May 2011 with a total number of 20529 measurements. The data exhibits a wide variety of environmental events that include short-term events such as rainfall as well as long term changes in measurement related to seasonal effects.

B. Parameter Settings

MoPBAS methods consist of a multitude of tuneable parameters, which can be used to control the sensitivity of the anomaly detection process. To obtain an optimized set of parameters for MoPBAS, the standards training, evaluation and testing procedure needs to be carried out. However, a fully annotated dataset, which is required for this process, is not available at the time of this paper is written. Thus, the initial parameter values used in these experiments are set based on the nature of the environment and the knowledge gained from on-site observation and site surveys. In our experiments, the following values were used for salinity anomaly detection:

- $N = 24$: N is the number of elements of the trend model B. Increasing N will reduce the sensitivity of the system as there is high probability that there might be an element that is close to the incoming sensor reading. However, only the normal values will be pushed into the trend model, thus further increases in N only duplicates existing elements (elements in the trend model are similar to each other). N is set to 24 (6 hours with 15 minutes sampling interval) in the following experiments. This is based on the average duration of change from high to low water or visa versa.
- $T_{inc/dec} = 0.02$: The step of the threshold T increases or decreases. Detection performance is not very sensitive to this value and this value is increased if the data exhibit a high degree of variability . This value depends on three main factors, the duration of an event, sampling rate and how fast sensor readings stabilise after an event. The number of $T_{inc/dec}$ should allow an increase of T from minimum to maximum longer than events and roughly the same length as the time required for stabilisation. Setting $T_{inc/dec}$ to 0.02 gives 70 steps from T_{upper} to T_{lower} giving approximately a 17 hour stabilisation period.
- $T_{upper} = 12$: The upper bound of the decision threshold. Increasing this value will reduce the sensitivity of anomaly detection, i.e., only large variations will be classified as anomalies. This value depends on the average of sensor measurements at the site and how the user defines an outlier. At our estuarine pilot site, the mean salinity value is 30.2 ppt over the length of the test dataset described. Readings are generally stable and we define any sudden changes with a magnitude greater than 2.5 ppt as an outlier. Thus, by mapping these values to the distance calculation function, T_{upper} can be calculated:

$$(T_{upper} \approx \sqrt{|(Sal_{average} \pm 2.5)^2 - Sal^2_{average}|}).$$

- $T_{lower} = 3$: The lower bound of the decision threshold. Reducing this value will increase the sensitivity of anomaly detection, smaller changes will be classified as an anomaly. Similar to T_{upper}, any salinity changes less than ± 0.15 ppt are considered as noise. Thus T_{lower} can be calculated from the same equation above.
- $T_{scale} = 3$: This is the equilibrium factor, which stretches $\overline{d}_{min}(t)$ to the same range as the threshold. Figure 6 shows the distribution of salinity background dynamics $\overline{d}_{min}(t)$. It appears that most of the $\overline{d}_{min}(t)$ values are less than 1 (this variation is generally attributed to sensor measurement error). Thus, to scale the $\overline{d}_{min}(t)$ to the same range of T, T_{scale} is set to 3 ($T_{lower}/1$)
- $L_{inc} = 5$: This is the trend model learning rate control parameter R increasing interval. Figure 6 and Figure 7 show that most of the $\overline{d}_{min}(t)$ values of salinity and turbidity are smaller than 2 and 3, respectively, thus,

we set this value to 5, which is large enough to give a rapid decrease in the learning rate when an significant event occurs.

- $L_{dec} = 0.1$: This is the trend model learning rate control parameter R decreasing interval. The value taken depends on the distribution of the background trend dynamic. Thus, the $\overline{d}_{min}(t)$ varies between 0 and 3 when no events are happening. The chosen value of 0.1 results in a smooth increase in the trend model updating probability.
- $R_{upper} = 3$: The upper bound of learning rate control parameter R. The value taken approximately equals the ratio between L_{inc} and the majority of $\overline{d}_{min}(t)$ values.
- $R_{lower} = 0.1$: The lower bound of the learning rate control parameter R. This takes the form of a small positive number to avoid negative probability. The ratio of R_{upper} and R_{lower} defines how fast the learning rate increases. For example, if $\overline{d}_{min}(t)$ is set to a constant, the learning rate will reach a maximum after 30 samples.

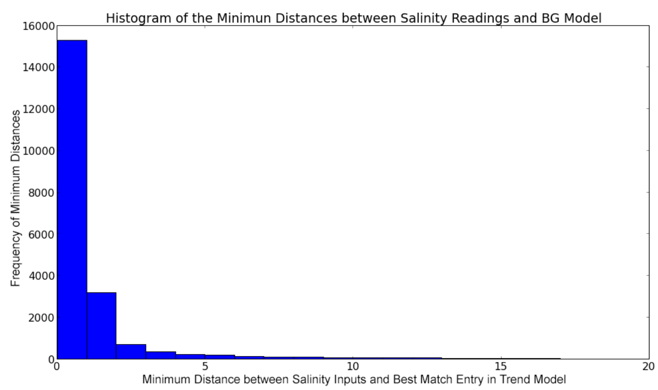


Figure 6. Histogram of salinity background dynamics

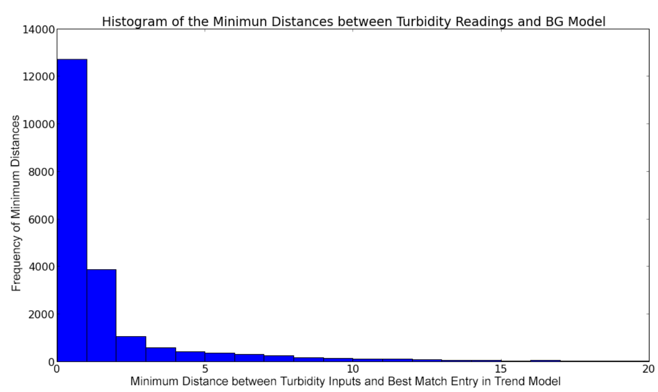


Figure 7. Histogram of turbidity background dynamics

As turbidity readings have different ranges and dynamics when compared with salinity measurements, the definition of a turbidity event are different. Some parameters need to be adjusted. The following parameter values were changed for turbidity anomaly detection: $T_{inc/dec} = 0.05$, $T_{scale} = 4.5$, $T_{upper} = 5$ and $T_{lower} = 1$. $T_{inc/dec}$ is modified because the stabilisation period of turbidity is generally shorter than salinity, which means that the decision threshold for turbidity needs to be updated faster. T_{upper} and T_{lower} are calculated using the same function. For turbidity, if the change is greater than $5NTU$, it will be considered as an anomaly and less than $1NTU$ will be considered as noise. However, the model updating control parameters, L_{inc}, L_{dec}, R_{upper} and R_{lower} remain the same. The number of elements of the trend model B is also set to 24.

For the event construction and clustering purposes, the parameters are the same for both salinity and turbidity. The number of words in the codebook is set to 50. When constructing an event, T_{gap} is set to 1 to avoid noise. This means that two anomalies are merged into the same event if the gap between them is smaller than 2 samples.

There are a number of events that may cause rapid changes in sensor readings, for example rainfall events, flood events, shipping or contamination events. In the present work, the number of clusters is set to 14, which is chosen as it represents approximately twice the number of known events that may occur at the testing site. However, the number of the cluster centres is application dependant. Increasing this number will further spilt the cluster into smaller sub groups.

C. Salinity Experiment Results

Applying the described MoPBAS anomaly detection to our test dataset resulted in 947 out of 20529 measurements being detected as anomalies. Figure 8 shows a 10-day window of the anomaly detection results. The red dots indicate salinity anomalies detected, while the blue line is the sensor measurements and the green solid line is the closest matching entry in the background trend model. As illustrated in Figure 8, most of the abnormal salinity readings are detected accurately.

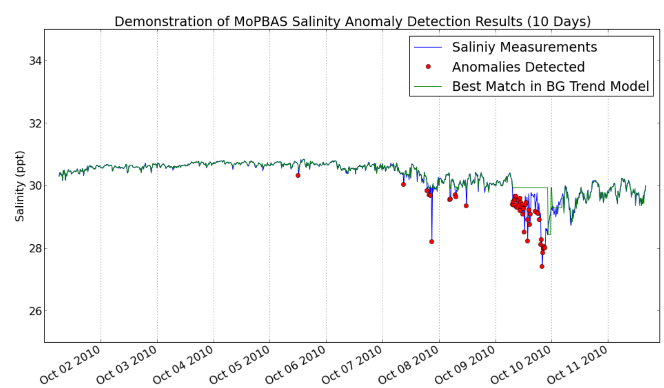


Figure 8. A 10-day window of the MoPBAS salinity anomaly detection results.

Figure 9 demonstrates adaptation of the detection threshold and background learning rate based on variation in the mean minimum distance ($\overline{d}_{min}$) between sensor measurements and background trend model. The red line at the bottom represents the background learning ratio. The decision threshold is shown in blue and the minimum distance between sensor readings and the best match entry in the model is shown in green.

In order to cluster events into groups based on their similarity, detected anomalies are merged into events based on

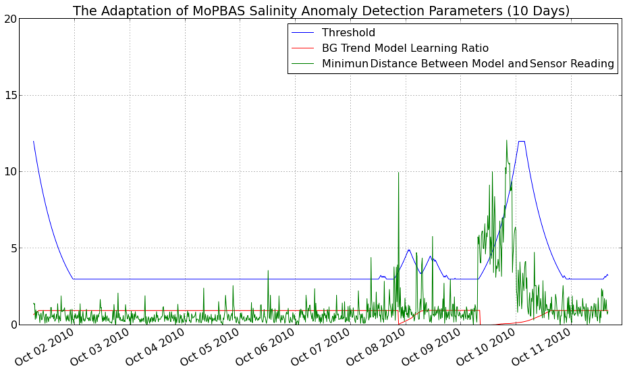


Figure 9. Detection threshold, background trend model learning rate and minimum distance between input value and best match element in model.

timestamps. From the 947 outliers detected, 261 events were constructed in the test datasets using the MoPBAS method. For each salinity anomaly detected, a set of features is extracted as the feature vector of the sample. To normalise these features, a 50-word codebook is created. Due to the limited dataset available, the code book is built using all anomalies. However, when more data is collected the codebook can be reused and does not need to be rebuilt unless the setup of the deployed system is modified or the monitoring site is greatly changed. Thus, each anomalous value is normalised as a "word" and the histogram of the occurrence of each word for each event constructed is used as the feature set of the event.

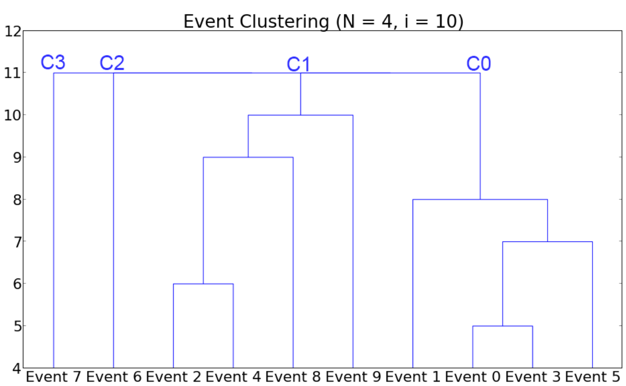


Figure 10. Example of how robust on-line clustering performs, note that for illustration purposes the number of cluster centres is set to 4 (N = 4) and the first 10 events (i = 10) are used.

TABLE I. CLUSTERING RESULTS, SHOWING THE NUMBER OF SIMILAR EVENTS WITHIN EACH CLUSTER GROUP.

Clusters	Number of events	Events
Clusters 0	243	Event 0, 1, 19, 22 etc.
Clusters 1	4	Event 57, 90, 197, 211 etc.
Clusters 2	2	Event 7, 101
Clusters 3	2	Event 204, 256
Clusters 4	2	Event 132, 142
Clusters 5	1	Event 10
Clusters 6	1	Event 34
Clusters 7	1	Event 43
Clusters 8	1	Event 62
Clusters 9	1	Event 80
Clusters 10	1	Event 91
Clusters 11	1	Event 157
Clusters 12	1	Event 173
Clusters 13	1	Event 180

Figure 10 illustrates how similar events are merged into the same cluster. At step 4 ($i = 4$), the cluster buffer is full. At step 5, the two most similar events, event 0 and 3 are merged. Event 2 and event 4 are grouped together when event 5 occurs. As can be seen from the graph, when event 9 occurs it is assigned to cluster 1, other events in cluster 1 (events 2, 4, 8) are the similar events to event 9, which happened in the past.

After applying the described clustering methods to the whole dataset, a total of 14 clusters are created. The number of events in each cluster is shown in Table I. Cluster 0 contains the most number of events. Cluster 1 consists of 4 similar events and there are 2 events in cluster 2, 3, and 4. Cluster 5 to cluster 13 only have 1 event in each.

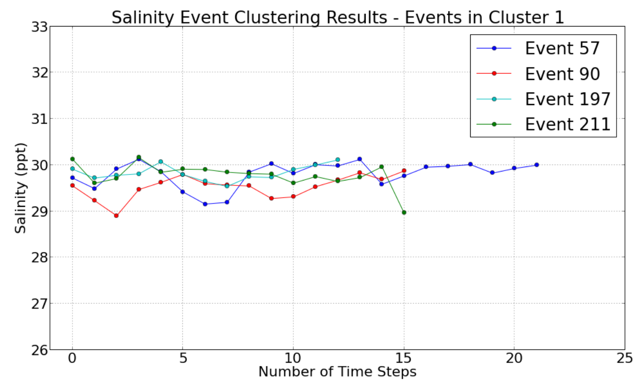


Figure 11. Plot of the salinity measurements of the four events in cluster 1.

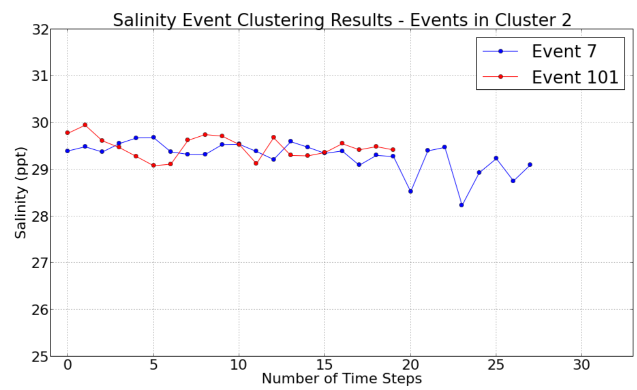


Figure 12. Cluster 2 consists of two events (event 7 and event 101).

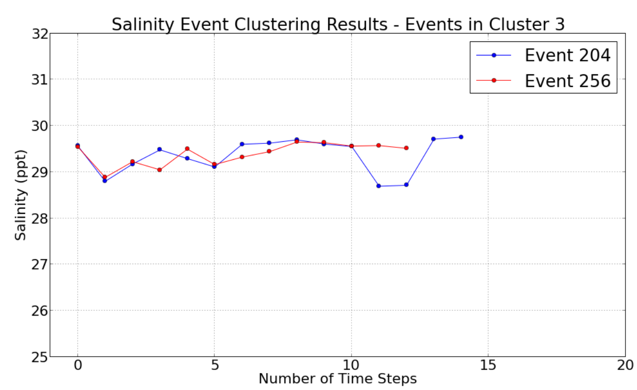


Figure 13. Two events (event 204 and event 256) in cluster 3.

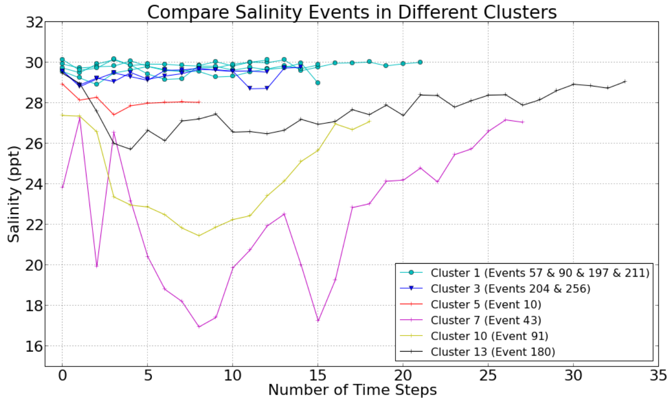


Figure 14. Comparison of salinity events in different clusters.

Figure 11 demonstrates the four events in cluster 1, the results show that events are similar to each other within the cluster. Figure 12 shows all the events in cluster 2 where it can be see that two events do have similar variations. Figure 13 demonstrates events 204 and 256 contained within cluster 3. It can be seen that the two salinity events have a very similar trend until the last two measurements, where event 256 has a small concave but event 204 remain flat. Figure 14 illustrates the difference between events in different clusters. As can be seen from the graph, events within the same cluster have similar trends but events in different clusters have very different profiles.

D. Turbidity Experiments Results

Applying the same procedures to turbidity data, 2096 sensor measurements are classified as anomalies. Figure 15 demonstrates a 10-day subset of turbidity anomaly detection results. The red dots are the turbidity anomalies detected, blue line is the sensor measurements and the green solid line is the closest matching entry in the background trend model. As illustrated in Figure 15, the majority of the abnormal turbidity readings are detected accurately.

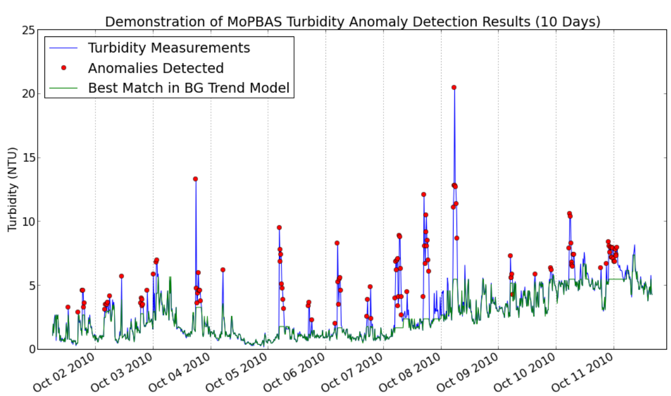


Figure 15. A 10-day window of the MoPBAS turbidity anomaly Detection Results.

Figure 16 demonstrates adaptation of the detection threshold and background learning rate based on variation in the mean minimum distance ($\overline{d}_{min}$) between sensor measurements and background trend model. As with detection of anomalies in the salinity dataset, the classification threshold increases when readings become highly variable and decreases when measurements do not change rapidly. In contrast, the model learning rate decreases sharply when events happen and increases slowly when sensor readings are stable.

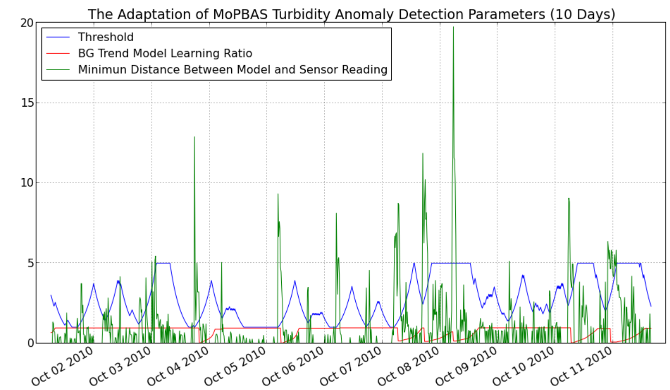


Figure 16. Detection threshold, background trend model learning rate and minimum distance between input value and best matching element in the background model.

Turbidity anomalies are grouped into events according to their timestamps. For the whole dataset, 707 events are constructed from a total number of 2096 classified anomalies. Table II lists the clustering results and the turbidity events in each cluster.

TABLE II. RESULTS OF TURBIDITY EVENT CLUSTERING, SHOWING THE NUMBER OF SIMILAR EVENTS IN EACH CLUSTER GROUP.

Clusters	Number of Turbidity Events	Events
Clusters 0	691	Event 0, 1, 10, 21, 152 etc.
Clusters 1	4	Event 71, 644, 647, 697
Clusters 2	2	Event 124, 253
Clusters 3	1	Event 29
Clusters 4	1	Event 99
Clusters 5	1	Event 102
Clusters 6	1	Event 252
Clusters 7	1	Event 570
Clusters 8	1	Event 606
Clusters 9	1	Event 608
Clusters 10	1	Event 610
Clusters 11	1	Event 705
Clusters 12	1	Event 706
Clusters 13	1	Event 707

Figures 17 and 18 show all events in the corresponding cluster where it can be seen that the events within the same cluster have similar variations. Three out of four events in cluster 1 have a spike shape at the beginning and then settle down. Although, the event 697 does not have a spike shape at the start but its overall trend is very similar to the settle down period of the rest of the events in the cluster. As can be seen in Figure 18, the two events are different in length. This shows the advantage of bag-of-words approach, which encode anomaly features as constant length descriptors. Figures 19 and 20 show that there is only 1 event in each cluster. However, as illustrated in the figures, the two events have unique trends.

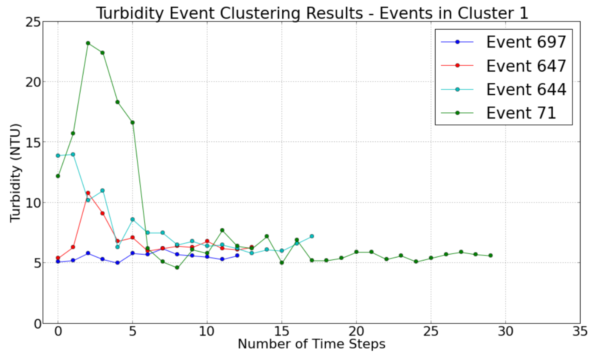


Figure 17. Plot of the turbidity measurements arising from events classified as being in cluster 1.

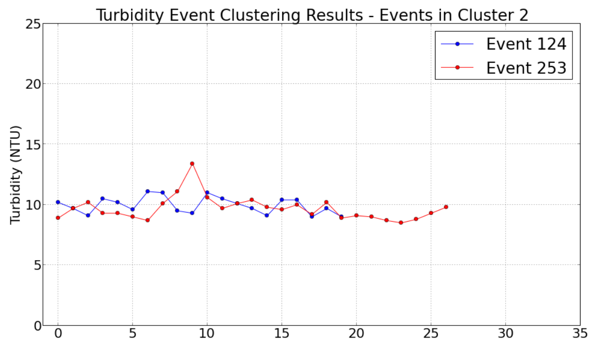


Figure 18. Plot of the turbidity measurements from events in cluster 2.

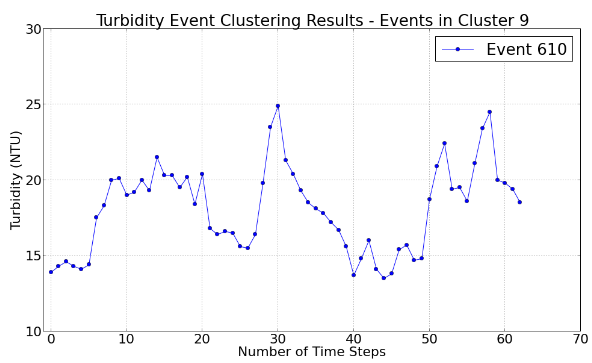


Figure 19. Plot of the turbidity measurements from events in cluster 9.

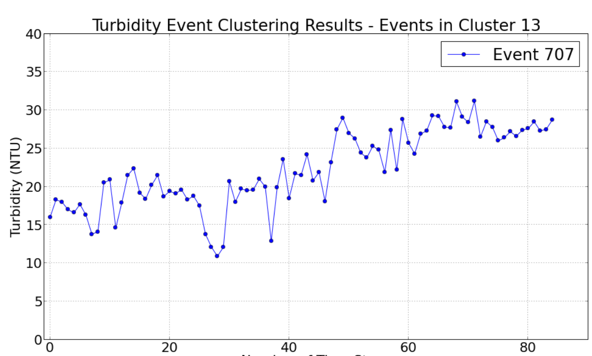


Figure 20. Plot of the turbidity measurements from events in cluster 13.

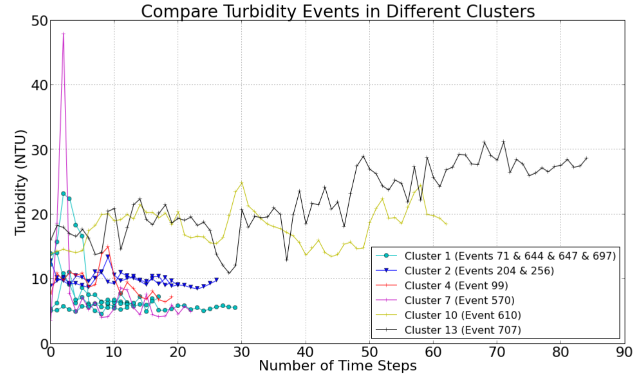


Figure 21. Illustration of the differences in turbidity readings between assigned clusters.

Figure 21 illustrates the difference between events in different clusters. As can be seen from the graph, events within the same cluster have similar trends but events in different clusters have very different shapes.

E. Discussion

We have shown, using time series of both salinity and turbidity, that the MoPBAS method is suitable for detection of anomalous sensor readings. Updating the background trend model provides the capacity to process both highly variable data and gradual changes such as tide or seasonal effects. The dynamic threshold and model update rate are appropriate for detection of environmental events in estuaries. As can be seen from Figures 9 and 16, the classification threshold is increased when an anomaly is detected. This is due to the fact that after an event occurs, there is usually a period of time where the water body settles down. The *in situ* sensor measurements is likely return to a similar trend or slightly offset compare to sensor readings prior the occurrence of the event. During this period, the sensor readings are typically highly variable and alter in step changes rather than monotonic decreases. The rapid increasing of the threshold during a event and slowly decreasing after the event can reduce false positives. Another advantage of use of this adaptive threshold is that the system only detects large variations during periods of high fluctuation while small changes will be captured during periods of relative stability. In contrast, the background learning rate remains high during stable periods and decreases rapidly when an anomaly is detected. This is because the background model should simulate the trend of the water quality parameters. However, as the threshold is raised the input is likely to be classified as normal even it is relatively different to the average trend. So the model learning rate is increased and the trend model will be updated as soon as the sensor readings are returning to normal. One of the benefits of the proposed MoPBAS method is ease of computation, meaning that that the process can be potentially performed on site or event on chip. This provides the opportunity to reduce the data transmission requirement, in which only the information on anomalous events will be sent back to a base station to enable a real-time alert system and normal readings can be logged locally or discarded. This could be a key factor for monitoring sites where the cost of data transmission is very high. Moreover, data transmission over long distance always consumes the majority of power in

WSNs. Thus, applying anomalous detection on site or on chip can extends the deployment time of WSNs which are battery dependant. This novel anomaly detection method inspired from image processing domain provides the fundamental block of creating a dynamic smart wireless sensor network. Above all, it appears that MoPBAS is an suitable anomaly detection approach for wireless sensor networks in the marine environment.

Results in Figures 14 and 21 show that the clustering successfully discriminates between events, assigning them to clusters where events within the same cluster are relativity similar to each other. Unique events are treated as new cluster centres (such as clusters 9 and 13 in the turbidity clustering example). This is a key advantage of using ROC clustering method compare to other commonly used techniques. These unique events are assigned as new cluster centres by ROC, rather than noise in some methods such as K-Means. This feature is very important from a water quality event detection perspective as these events have no analogous events in the past, and thus may be potentially of greater importance to operators. These are the events that would then trigger an alert when detected, thus allowing operators to react accordingly. However, further analysis and determination of the causes and effects of these events require fusion of information from multiple data sources as proposed in the smart sensing system in Figure 1. Another advantage of ROC method is that it is computational inexpensive, which the classification of abnormal events and the update of the model can be performed in real-time. In addition, ROC method is easy to interpret and its tree-like structure can can used to build an event indexing and retrieving system. Defining the number of centres K in ROC is crucial. In this paper, we assigned K equal to 14, which is based on our site surveys and our assumption of the number of possible abnormal event types that may occur at our pilot site. However, finding a suitable K for marine environmental monitoring requires large amount of data collected over long time periods due to seasonal and weather effects.

VI. Conclusion

In this paper, we have demonstrated a novel system of detection and clustering of events in time series datasets of environmental turbidity and salinity measurements. We have modified the pixel-based adaptive segmentation technique from the image processing domain for this purpose and applied robust on-line clustering for classification of events. We have provided this in the context of a component of the proposed framework for an automated sensor network as part of a future smartbay-smartcity project. Such integrated *in situ* sensor networks for environmental applications have the potential to form a significant part of future smart city infrastructure. However, the data generated from such systems must be seamlessly integrated into the overall smart city-smart bay system if they are to be used to full advantage. Use of the generated data for automated data analytics including event detection and classification must be an integral part of this process if such systems are to become ubiquitous and useful in decision support. We altered a state-of-the art background modelling technique from the image processing domain to built a background trend model to detect anomalous events in commonly measured parameters from *in situ* sensor within estuarine systems such as conductivity (derived salinity) and turbidity, using real data generated with in the Dublin Smart City: Smart Bay Network. Furthermore, we have shown the utility of robust online clustering to classify detected events based on their characteristics. The root environmental causes of these events can now be ascertained and a significance level assigned to these events (for example pollution, human activity or storm events). We have demonstrated the utility of this approach in Dublin Bay for detection of abnormal events, and the potential of these techniques for classifying and further enhancing decision support within this framework. By combining the outcome of other parallel research work such as ship traffic detection from visual sensor, the proposed novel multi modality smart sensing system can potentially provide a rich content based indexing and retrieval system to assist the marine scientists better and easier understand and modelling the marine ecosystem. Subsequently, the system can support decision makers in construction of new policies to better protect environmental and coastal resources.

Acknowledgments

The authors acknowledge support from Science Foundation Ireland (grants 07/CE/I147 and SFI/12/RC/2289) and the Beaufort Marine Research Award, carried out under the Sea Change Strategy and the Strategy for Science Technology and Innovation (20062013), with the support of the Marine Institute, funded under the Marine Research Sub-Programme of the National Development Plan 2007-2013.

References

[1] F. Regan, D. Zhang, T. Sullivan, C. Briciu, H. Cooney, K. Murphy, E. O'Connor, N. O'Connor, and A. Smeaton, "A smart city-smart bay project-establishing an integrated water monitoring system for decision support in dublin bay," in *SENSORCOMM 2013, The Seventh International Conference on Sensor Technologies and Applications*, 2013, pp. 75–82.

[2] R. Greenwood, J. Webster, and F. Regan, "Sustainable water: Chemical science priorities. chapter 4 - water monitoring," Royal Society of Chemistry, Tech. Rep., 2007.

[3] J. Grath, R. Ward, A. Scheidleder, and P. Quevauviller, "Report on eu guidance on groundwater monitoring developed under the common implementation strategy of the water framework directive," *J. Environ. Monit.*, vol. 9, pp. 1162–1175, 2007.

[4] A. Lawlor, J. Torres, B. O'Flynn, J. Wallace, and F. Regan, "Deploy: a long term deployment of a water quality sensor monitoring system," *Sensor Review*, vol. 32, no. 1, pp. 29–38, 2012.

[5] B. O'Flynn, F. Regan, A. Lawlor, J. Wallace, J. Torres, and C. O'Mathuna, "Experiences and recommendations in deploying a real-time, water quality monitoring system," *Measurement Science and Technology*, vol. 21, no. 12, p. 124004, 2010.

[6] C. Albaladejo, P. Sánchez, A. Iborra, F. Soto, J. A. López, and R. Torres, "Wireless sensor networks for oceanographic monitoring: A systematic review," *Sensors*, vol. 10, no. 7, pp. 6948–6968, 2010.

[7] P. W. Rundel, E. A. Graham, M. F. Allen, J. C. Fisher, and T. C. Harmon, "Environmental sensor networks in ecological research," *New Phytologist*, vol. 182, no. 3, pp. 589–607, 2009.

[8] J. Ni, C. Zhang, L. Ren, and S. X. Yang, "Abrupt event monitoring for water environment system based on kpca and svm," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 4, pp. 980–989, 2012.

[9] D. Hart, S. A. McKenna, K. Klise, V. Cruz, and M. Wilson, "Canary: a water quality event detection algorithm development tool," in *Proceedings of the World Environmental and Water Resources Congress, ASCE, Reston, VA*, 2007, pp. 1–9.

[10] E. Vugrin, S. A. McKenna, and D. Hart, "Trajectory clustering approach for reducing water quality event false alarms," in *Proc., of ASCE Annual World Environmental and Water Resources Congress, Kansas City, Missouri, May*, 2009, pp. 17–21.

[11] S. A. McKenna, E. D. Vugrin, D. B. Hart, and R. Aumer, "Multivariate trajectory clustering for false positive reduction in online event detection," *Journal of Water Resources Planning and Management*, vol. 139, no. 1, pp. 3–12, 2012.

[12] L. Perelman, J. Arad, M. Housh, and A. Ostfeld, "Event detection in water distribution systems from multivariate water quality time series," *Environmental Science & Technology*, vol. 46, no. 15, pp. 8212–8219, 2012.

[13] S. A. McKenna, D. Hart, K. Klise, V. Cruz, and M. Wilson, "Event detection from water quality time series," in *Proc. ASCE World Envir. & Water Resources Congress, Tampa, Fla*, 2007.

[14] K. P. Singh, N. Basant, and S. Gupta, "Support vector machines in water quality management," *Analytica Chimica Acta*, vol. 703, no. 2, pp. 152–162, 2011.

[15] K. P. Singh, A. Basant, A. Malik, and G. Jain, "Artificial neural network modeling of the river water qualitya case study," *Ecological Modelling*, vol. 220, no. 6, pp. 888–895, 2009.

[16] H. R. Maier and G. C. Dandy, "The use of artificial neural networks for the prediction of water quality parameters," *Water Resources Research*, vol. 32, no. 4, pp. 1013–1022, 1996.

[17] W. Wu, G. C. Dandy, and H. R. Maier, "Protocol for developing ann models and its application to the assessment of the quality of the ann model development process in drinking water quality modelling," *Environmental Modelling & Software*, vol. 54, pp. 108–127, 2014.

[18] S. A. McKenna, M. Wilson, and K. A. Klise, "Detecting changes in water quality data." *Journal: American Water Works Association*, vol. 100, no. 1, 2008.

[19] M. W. Koch and S. A. McKenna, "Distributed sensor fusion in water quality event detection," *Journal of Water Resources Planning and Management*, vol. 137, no. 1, pp. 10–19, 2010.

[20] D. Hou, H. He, P. Huang, G. Zhang, and H. Loaiciga, "Detection of water-quality contamination events based on multi-sensor fusion using an extented dempster–shafer method," *Measurement Science and Technology*, vol. 24, no. 5, p. 055801, 2013.

[21] D. Ömer Faruk, "A hybrid neural network and arima model for water quality time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 586–594, 2010.

[22] M. V. Storey, B. van der Gaag, and B. P. Burns, "Advances in on-line drinking water quality monitoring and early warning systems," *Water Research*, vol. 45, no. 2, pp. 741–747, 2011.

[23] F. Regan, A. Lawler, and A. McCarthy, *SmartCoast Project- Smart Water Quality Monitoring System*. Environmental Protection Agency, Ireland, 2009.

[24] F. Regan, A. Lawlor, B. OFlynn, and J. Wallace, "Deploy: Smart demonstration of online water quality monitoring on the river lee, cork, ireland," Environmental Protection Agency, Tech. Rep., 2011.

[25] S. Roth and J. G. Wilson, "Functional analysis by trophic guilds of macrobenthic community structure in dublin bay, ireland," *Journal of Experimental Marine Biology and Ecology*, vol. 222, no. 1, pp. 195 – 217, 1998.

[26] J. Wilson, "Productivity, fisheries and aquaculture in temperate estuaries," *Estuarine, Coastal and Shelf Science*, vol. 55, no. 6, pp. 953 – 967, 2002.

[27] DCC, *Dublin to Waste Project. Environmental Impact Statement*, D. C. Council, Ed. Dublin City Council, 2006, no. 246851, chapter 12.

[28] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, pp. 38–43.

[29] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[30] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval," *Pattern Recognition*, vol. 35, no. 8, pp. 1675–1686, 2002.

[31] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, June 2005, pp. 524–531 vol. 2.

[32] D. Zhang, S. Chen, and K. Tan, "Improving the robustness of online agglomerative clustering methodbased on kernel-induce distance measures," *Neural Processing Letters*, vol. 21, no. 1, pp. 45–51, 2005.

International Journal on Advances in Intelligent Systems, vol 7 no 1 & 2, year 2014, http://www.iariajournals.org/intelligent_systems/

A Spaces-Based Platform Enabling Responsive Environments

Daniela Micucci, Alessio Vertemati, Francesco Fiamberti, Diego Bernini, and Francesco Tisato

Department of Informatics, Systems and Communication
University of Milano - Bicocca
Milano, Italy
Email: {daniela.micucci, alessio.vertemati, francesco.fiamberti, diego.bernini, francesco.tisato}@disco.unimib.it

Abstract—**Space Integration Services is a software communication platform that enables the seamless integration of sensors, actuators, and application-logic components through a multi-space model and a spaces-based publish/subscribe mechanism. The underlying model is based on finite spaces only, thus the application-logic components that need to reason on spaces like the geodetic or the Cartesian ones, are obliged to use a discretized version of those spaces and to maintain the correspondence between "real" locations and their discretized representation if they are interested in exploiting the services offered by the platform. To compensate for these limitations, the platform has been extended in order to add support for spaces with an infinite number of locations, (e.g., spaces described by continuous coordinates, such as geodetic or Cartesian spaces). Such extension is essential for the use of SIS also as enabling platform for outdoor pervasive computing systems thanks also to the wide spread of outdoor localization systems. This paper presents the new conceptual model that results from a generalization of the existing one valid for finite spaces only. To encompass both finite and infinite spaces, the new model moves the primitive concept of elementary position from *location* to *spatial context* (as a set of locations). Several kind of spatial contexts have been introduced to offer a (quite) complete set of localization typologies, for example, from a single position to a set of positions described by a function. The model has been turned in a prototypal implementation realized by means of an additional layer on top of the finite-space version. Such an implementation has been experimented in a real case scenario dealing with a parcel distribution company. Finally, the performance of the prototype is then compared to the one of the finite-space version in a series of experimental tests.**

Keywords–infinite spatial models; spaces-based communication; software architecture; location-aware; responsive environments.

I. INTRODUCTION

Space Integration Services (SIS) [1] is a platform that enables pervasive computing [2], supporting information flows between devices and applications. Pervasive computing aims at simplifying the everyday life through digital environments that are sensitive, adaptive, and responsive to the user's needs. A pervasive computing system requires the perception of the context in which the user operates to provide a richer and expanded mode of interaction, in addition to an intelligence for performing actions on the environment. From a technological point of view, pervasive computing relies on responsive environments. The term responsive environment [3] refers to physical environments enhanced by input devices (e.g., sensors or cameras) and output devices (e.g., displays, lights, motors). Input devices capture stimuli from the environment, whereas output devices execute actions on the environment given a predefined set of commands.

Responsive environments are, therefore, able to perceive and respond to users thanks to the presence of a computer system that receives data from the sensors (input stream) and sends commands to the actuators (output stream). For example, an application may locate users onto the cartographic representation of the city and may also receive data from light sensors (input stream) about their state (e.g., on and off). On the basis of established rules, an application could send commands to the street lights (output stream). A rule can state that in nighttime, if a user is close to a street light (within a distance of ten meters), the street light should be turned on. When no one is close to the street light, then the street light should be turned off. This simple example emphasizes how responsive environments require establishing information flows among the devices and the applications. SIS fulfills the above requirement by enabling a seamless integration of sensors, actuators, and application-logic components through a multi-space model and a spaces-based publish/subscribe mechanism. It provides various spatial models that can be used by applications to represent location-related information in order to support complex representations of the environment with a focus on location-aware systems. In this regard, different spatial representations can be put in appropriate correspondence to describe the localization according to different visions of the environment. This allows devices and end-user applications to reason on an own representation of the environment and thus are not obliged to share a common representation. For example, an application that localizes persons inside a building reason in terms of rooms and passages among them, whereas the devices reason in terms of a Cartesian representation of the controlled area. From the knowledge of the authors, platforms offering interoperability rely on a shared view of the space. Moreover, most of them provide spatial representations of the physical environment only. On the opposite, SIS allows to model any kind of space, be it physical (e.g., a geographical area) or logical (e.g., an organization chart).

SIS originates from a preliminary version that supported finite spaces only (symbolic models and grid models) [4]. Such kind models, even if discrete, were nevertheless sufficient to create indoor pervasive computing applications. For example, switching on and off of lights depending on the presence of people in the rooms. Recognition of people and then setup the environment on the basis of their optimal configurations, and so on. With the increase of accuracy and precision of localization sensors [5] and with the need to include geographical-related spatial models to support outdoor applications also, a revision of the platform was required. The revision aims at introducing mechanisms to deal with spatial representations that contain a potentially infinite number of locations, for example those described by continuous coordinates, such as geodetic and Cartesian spaces, but also unbounded grids described by discrete coordinates. For example, such a platform will support applications that require to locate persons outdoor (they need to reason on a geographical space) or that require to define the fine-grained trajectory of a mobile robot (they need to reason on an Euclidean space).

Finally, the new version of SIS was designed from the previous one in order to reuse the base mechanisms that support information flows and that constitute the core of the platform. Those mechanisms, which are based on finite spaces, are then very efficient. The idea was then to yield the same mechanisms, properly managing flows between infinite spaces as if they were flows between finite spaces.

The main contributions of this article are a deep description of both the conceptual model and the platform reifying it. Moreover, since the implementation has been completely revised and completed with respect to the results presented in [1], some of its interesting issues will be discussed. Finally, tests have been reformulated in order to consider different configurations with different hardware.

The paper is organized as follows. Section II presents the overall SIS conceptual model. Section III summarizes the principal services that SIS should provide to applications. Section IV discusses the use of the model in an application scenario. Section V presents some issues related to the actual implementation of the platform. Section VI presents the results of several tests aimed at estimating the performance of the proposed extension with respect to the existing SIS implementation considering different configurations with different hardware. Section VII reviews related works. Finally, conclusions and future developments are presented in Section VIII.

II. CONCEPTUAL MODEL

The conceptual model of the finite space-based SIS model was based on the assumption that *spaces* are finite sets of *locations* built from *spatial models* (e.g., graph spatial model, name spatial model, and grid spatial model). Non- empty sets of locations belonging to a space are named *spatial contexts*. In such a model, locations play a crucial role since the existence of both spaces and contexts is subject to the existence of the set of locations that constitutes them. In other words, spaces and spatial contexts are defined as the collection of all the locations that constitute them. This structural constraints is clearly impossible to fulfill when dealing with infinite spaces, that is, spaces that contain a potentially infinite number of locations. Thus, the conceptual model has been revised around a new definition of spatial context that becomes the elementary localization building block.

A. Space and Spatial Model

A *space* is a set of *potential location*s, that are all the locations that could be theoretically considered in that space. For example, if the Milan subway lines are described by means of a graph in which nodes model stops, then the potential locations are all the stops of all the lines. On the other hand, if a Cartesian space is used to localize entities within a room,

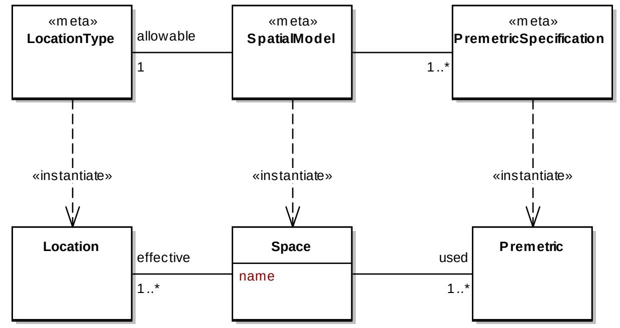


Figure 1: Core concepts: meta representation and corresponding instances

then the potential locations are every point in $\mathbb{R}^2$ of the area delimited by the room perimeter.

Applications, when dealing with a space, explicitly manage *effective location*s, which are a subset of space's potential locations. For example, an application that monitors the position of trains traveling on the Milan Red Line subway will only deal with locations that represent the stops on the Red Line. On the other hand, an application that calculates the trajectory of a mobile entity will only explicitly consider a finite number of locations in the Cartesian space, that is, the locations belonging to the trajectory.

Spaces and effective locations are respectively built from spatial models and location types. A *spatial model* specifies the type of *allowable location*s (*location type*), the way in which locations are arranged, and at least one premetric that can be applied to a pair of locations (*premetric specification*). The premetric defines the distance between two locations as a positive, non-zero number if the two locations are distinct, and zero if the locations are the same. A *location type* specifies the structure of potential locations (e.g., a couple of double values).

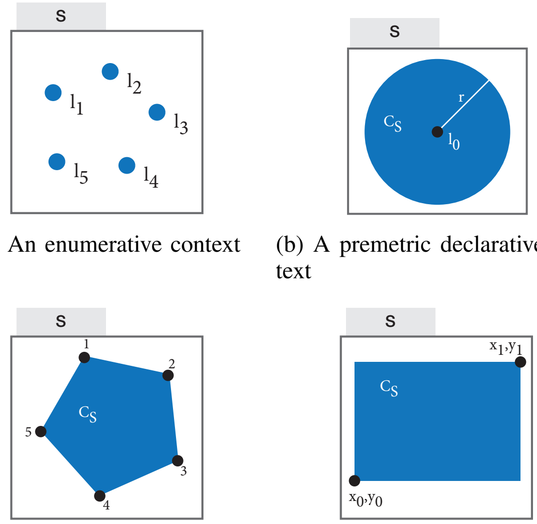
The key aspect, shown in Figure 1, is that location types, spatial models, and premetric specifications are meta-level concepts (≪meta≫ stereotype) because, according to them, it is possible to instantiate locations, spaces, and premetrics respectively, that are base-level concepts (i.e., they are the element an application manages when dealing with localization issues). The direction of arrows in Figure 1 emphasizes that base elements originate from meta elements. In an object oriented language, it is possible to compare meta-level elements to classes and base-level elements to objects instantiated from classes. In other words, space and spatial model, location and location type, premetric and premetric specification respectively belong to different levels of abstraction.

The definition of space applies both to cases in which the set of locations is infinite and to cases in which it is finite. In fact, with the term *potential* locations we refer to all the locations that can be defined according to the spatial model. Therefore, we distinguish potential locations from effective locations that are the actual ones used by applications.

Two spatial models representing spaces with potentially infinite number of locations have been defined: the geodetic and the n-dimensional Cartesian. The geodetic spatial model defines locations as geodetic coordinates (i.e., latitude and longitude) in a geodetic coordinate reference system [6]. An applicable premetric is the orthodromic distance [7], which is the shortest distance between two points expressed by geodetic coordinates on the surface of a sphere. The orthodromic distance is also a metric. The n-dimensional Cartesian spatial model represents an Euclidean space $\mathbb{R}^n$. In this model, locations are represented by ordered tuples of real numbers in an orthogonal Cartesian reference system. A premetric for the Cartesian spatial model is the Euclidean distance, which is also a metric.

These examples highlight that a location in an infinite space can be identified with a potentially infinite precision, as in the nature of the coordinate system based on real numbers.

The spatial models defined in the previous finite space-based SIS model (i.e., graph, grid, and name spatial models) are declinable as special cases of the new definition of space in which the allowable locations are finite in number and are represented with finite precision. For example, in a grid space, allowable locations are the cells within the limits of the grid and are represented by their respective indexes defined on the set of integers.



(a) An enumerative context

(b) A premetric declarative context

(c) A polygonal context

(d) A pure functional context

Figure 2: Different types of spatial context

B. Spatial Context

In the finite space-based SIS model spatial contexts have been introduced in order to handle the selection of subsets of locations. The definition of spatial context has been refined to be also applicable in spaces with potentially infinite number of locations.

A *spatial context* C_S (simply *context* in the following) is a subset of potential locations of a space S. It is defined by a set of effective locations termed *characteristic locations* in S and by a *membership function* that states if a given location of S belongs to the context. Essentially, the membership function is a boolean function that is true when a location is in the spatial context. According to the membership function used, the following kinds of contexts have been identified: *enumerative*, *premetric declarative*, *polygonal*, and *pure functional*.

An *enumerative context* is a context in which the set of characteristic locations is non-empty and the membership function is based on the standard belonging relationship defined in set theory. The locations belonging to an enumerative context are identified through the enumeration of the characteristic locations. For example, given a space S, an example of enumerative context is defined by the set $C_S = \{l_1, l_2, ..., l_5\}$ of characteristic locations (see Figure 2a). For example, such kind of context can be useful to specify the room in which a person has been localized.

A *premetric declarative context* is a context in which the set of characteristic locations contains one location only and the membership function is a premetric one. Thus, the locations belonging to this kind of context are all the locations within a given distance from the characteristic location in terms of the premetric function (see Figure 2b). For example, such kind of context can be useful to specify the detection area of a RFID reader.

A *polygonal context* is a context in which the characteristic locations are the vertexes of a polygon, and the membership function indicates the inclusion of a location in the region corresponding to the polygon itself (see Figure 2c). The locations belonging to a polygonal context are all the locations within the area delimited by the polygon. For example, such kind of context can be useful to specify a delimited area inside a building or the field of view of a camera.

Finally, a *pure functional context* is a context in which the set of characteristic locations is empty and the membership function is defined by using mathematical expressions defined in terms of the space coordinate system. For example, consider a Cartesian bi-dimensional space S. A context can be defined by the following membership function: for all locations (x, y) in S and for any given location (x_0, y_0) and (x_1, y_1),

$$f(x,y) = \begin{cases} \text{true} & \text{if } x_0 < x < x_1 \text{ and } y_0 < y < y_1 \\ \text{false} & \text{otherwise} \end{cases} \quad (1)$$

(see Figure 2d). Obviously, this example aims at easily explaining the context idea. Indeed, it can be transformed into a polygonal one by interpolating the coordinates of the points that represent the extremes of the allowable values. Such a context should be used to face situations in which contexts are more complex and cannot be approximated with a polygonal one.

C. Projection

Related to the concept of space, is the concept of projection, which is widely used in geometry and in algebra.

The *projection function* allows the transformation of locations belonging to a space, called the source, in locations belonging to another space, called target. The target space can be defined according to the same spatial model of the source, or to a different one.

The transformation of the geodesic representation of the Earth to any other cartographic representation, the application

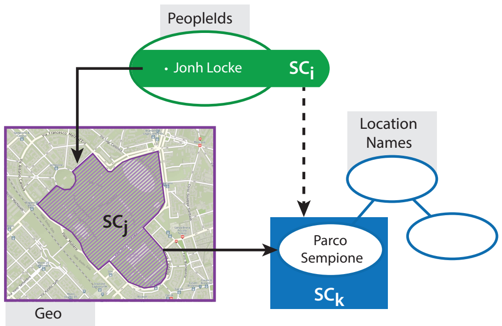


Figure 3: Explicit and implicit mappings

of a scale factor to a Cartesian representation, the transformation of a tri-dimensional environment onto a bi-dimensional surface, are all examples of the application of a projection function.

It is possible to observe how the application of the projection function to a region of the space may be useful in providing the representation of that region in agreement with the spatial model of the target space.

The projection function is used to define another kind of context termed *projective context*: given a source spatial context C_S defined in a source space S and given a target space T, the result of a projection is a spatial context C_T on the target space T containing the locations that are obtained by applying a projection function f to all the locations in the source context C_S.

D. Mapping

Mappings relate different spaces. For example, a mapping can relate an area of a Cartesian space (representing the plant of a building) to a node of a graph (representing a synoptic view of the same building). Mappings are key concepts because, as it will be explained afterward, they enable the communication among components, even if they rely on different spaces. Three kinds of mapping have been defined: *explicit*, *projective*, and *implicit*.

An *explicit mapping* is an ordered pair of contexts defined in different spaces (possibly based on different spatial models): given the two spaces S_1 and S_2 with $S_1 \neq S_2$, the ordered pair (SC_1, SC_2) is an explicit mapping between the contexts $SC_1 \subseteq S_1$ (source) and $SC_2 \subseteq S_2$ (target).

Figure 3 shows two examples of explicit mappings between the context SC_j of the *Geo* space (a geodetic spatial model) and the context SC_k of the *LocationNames* space (a graph spatial model), and between the context SC_i of the *PeopleIds* space (a name spatial model) and the context SC_j of *Geo*.

The target context may be defined independently of the source context. But when the target context is the result of the application of a projection to the source context, the mapping is termed *projective mapping* and is fully determined by the source context and the projection function.

Let SM be the set of all the defined explicit and projective mappings, and let SC_a and SC_b be contexts defined in different spaces. The *implicit mapping* (SC_a, SC_b)

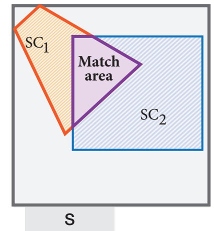


Figure 4: Direct matching

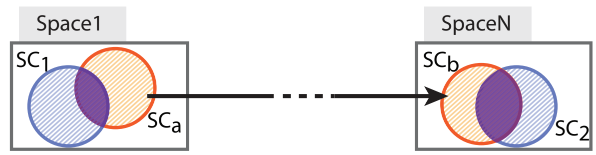


Figure 5: Indirect matching

is derived if there exist n contexts $SC_1, \ldots, SC_n$ such that $(SC_a, SC_1), (SC_1, SC_2), \ldots, (SC_n, SC_b) \in SM$ for $n >= 1$.

In Figure 3 the dotted arrow represents the implicit mapping (SC_i, SC_k) between the contexts SC_i of *PeopleIds* and SC_k of *LocationNames*. It is derived from (SC_i, SC_j) and (SC_j, SC_k) mappings.

E. Matching

A *direct matching* occurs when the intersection between two spatial contexts defined in the same space is not empty. For example, given a space S defined in $\mathbb{R}^2$ that models a room in a building, a pure functional context SC_1, and a polygonal context SC_2 both defined in S. A direct match occurs since the intersection between SC_1 and SC_2 is not empty (see Figure 4).

Let SC_1 and SC_2 be spatial contexts defined in different spaces. An *indirect match* between SC_1 and SC_2 occurs when there exists a mapping (explicit or implicit) (SC_a, SC_b) such that the intersections between SC_1 and SC_a and between SC_2 and SC_b are both not empty (see Figure 5).

III. Services

The concepts introduced in Section II are the basis of several services that SIS offers to applications. They can be divided into two main groups: 1) management and inspection of spaces' configurations; 2) communication support through spatial contexts (spaces-based publish/subscribe).

A. Space Structure Support

Before applications may reason on spatial contextualizations, spaces must be defined and properly configured. For this purpose, a set of services is provided.

When defining a space, applications must initially choose the appropriate spatial model (i.e., the spatial model that best fits the needed space). For example, if an application needs to plan the movements of a mobile entity inside a building in terms of sequences of rooms it has to traverse before reaching final destinations, then such an application may rely on a synoptic representation of the building built according to the graph spatial model; on the opposite, if another application is in charge of fine tuning the movement of the mobile entity, then it may rely on a Cartesian representation of the building built according to the Cartesian spatial model.

Once the spatial model has been chosen, it must be configured so that the resulting space will contain all the potential locations (i.e., all the locations an application possibly needs to use). For example, the synoptic view of the building can be obtained by specifying the list of nodes and arcs constituting the building; the Cartesian representation can be obtained by specifying the boundaries of the area containing the building.

From the above consideration, space definition is supported by the primitive

```
defSpace(SpaceName, Model, Parameters)
```

where `SpaceName` is the unique name for the new space, `Model` specifies the spatial model from which building the space, and `Parameters` are configuration information that depends on the spatial model. For example

```
defSpace(U14, GraphSpatialModel,
    <{r1, r2, r3}, {r1-r2, r1-r3}>)
```

defines a graph space named `U14` whose nodes are r_1, r_2, and r_3, and whose arcs are r_1-r_2 and r_1-r_3.

Once spaces exist, explicit mappings among spatial contexts can be specified through the

```
defMapping(Source, Target)
```

primitive, where `Source` and `Target` are respectively the source and the target contexts. Each context definition is specified by `<SpaceName, Structure>` where `SpaceName` is the unique name of the space in which the context is being defined, and `Structure` specifies the context typology, the set of characteristic locations, and the membership function. For example `<U14, <ENUM,` $\{r_1, r_2\}$, $\in$`>>` defines an enumerative context in the `U14` space whose characteristic locations are r_1 and r_2 and the membership function is the belonging function of the set theory. Suppose that a Cartesian space named `1St` has been defined representing the first floor of the U14 building, then

```
defMapping(
    <U14, <ENUM, {r1}, ∈>>,
    <1St, <POLYG, {p1, p2, p3, p4, p5}, f_in>>)
```

specifies a direct mapping. It may be interpreted as follows: room r_1 in U14 building is represented by the context `<U14, <ENUM,` $\{r_1\}$, $\in$`>>` in the graph space `U14` and by `<1St, <POLYG,` p_1, p_2, p_3, p_4, p_5, f_{in}`>>`, a polygonal context defined in the Cartesian space `1St`.

The above primitives can be used both when configuring SIS spaces before any application starts using the services, and during the normal execution of the applications. Moreover, SIS provides a set of primitives that allow applications to reason about spatial configurations. For examples:

- `getSpaces`: returns the list of all the defined spaces.
- `getSpaceInfo`: given the name of a space, returns all the related information (i.e., space typologies and parameters).
- `getMappings`: given two contexts, returns (if any) all the defined mappings.
- `getMapped`: given a context, returns (if any) all the defined mappings in which the context is involved.
- `getProjection`: given a context, a target space, and a projective function, returns the projected context.

The complete set of primitives that a platform reifying SIS model should provide is presented in the Figure 9 in Section V, which discusses the implementation we developed.

B. Communication Support

As previously introduced, SIS enables information flows relying on the publish and subscribe mechanism: applications *publish* information on spatial contexts, and *subscribe* on spatial contexts for the asynchronous reception of the published information. The published information is called *thematic information*, which is domain dependant and is not in any way interpreted by SIS.

Contexts of any kind may be defined on any spatial model: from the enumerative to the pure functional one. The choice of the context type depends on application domain issues. For example, in a situation of emergency, the State Forestry Department may report the area affected by a fire by publishing a thematic information (specifying information dealing with the fire) in a polygonal context (representing the affected area) defined in the geodesic space. On the opposite, if a room is equipped with a very accurate localization system (e.g., Ubisense), it may publish entity positions (the thematic information) exploiting enumerative contexts that contain the single position inside the Cartesian space modeling the room.

An application performs a publication via

```
publish(info, <C1, C2, ...>)
```

where `info` is the thematic information and `<`C_1, C_2, `...>` is a list of spatial contexts in which the information has been localized.

An application performs a subscription via

```
subscribe(C1, C2, ...)
```

where `<`C_1, C_2, `...>` is the list of spatial contexts in which the application declares its interests.

When an application performs a publication, the enclosed thematic information is received by all the applications that previously performed a subscription such that at least one of its contexts *matches*, either directly or indirectly, a context of the publication. The thematic information is enriched with all the contexts that contributed to the matching.

Referring to the example in Figure 6a, $\texttt{Component}_2$ subscribes to a context defined in the `Floor2D` space built from a Cartesian spatial model. The subscription context is polygonal (the hexagon in Figure 6a). $\texttt{Component}_1$ makes

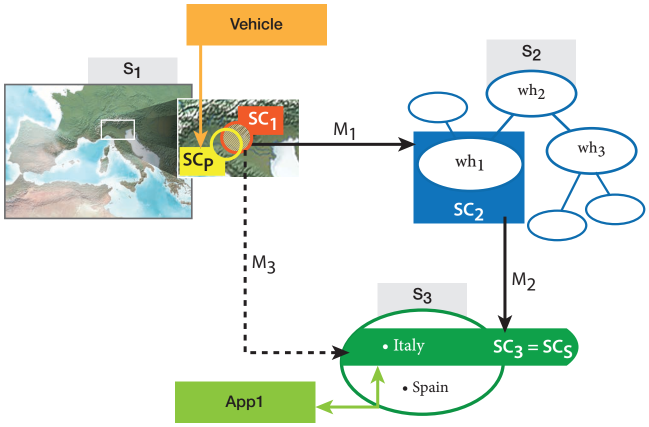


Figure 7: Example scenario

a publication on the same space Floor2D. The context is premetric (the circle in Figure 6a) and intersects the polygonal one. Thus, a direct match occurs and $\texttt{Component}_2$ receives the thematic information info enriched with additional information that includes, in addition to the time instant in which the match occurred, the complete contextualization, that is, all the contexts contributing the matching (the polygonal and the premetric).

In Figure 6b $\texttt{Component}_2$ and $\texttt{Component}_1$ reason on different spaces. $\texttt{Component}_2$ subscribes to an enumerative context (the highlighted node labeled "Living room" in Figure 6b) in the space FloorGraph space built from a graph spatial model. $\texttt{Component}_1$ makes the same publication as in 6a. Since a mapping between the publication and subscription contexts has been defined, an indirect match occurs so that $\texttt{Component}_2$ receives the thematic information info enriched with all the additional information (the matching time instant and the complete contextualization).

IV. Application Scenario

This section will apply the concepts introduced in both Section II and Section III considering an exemplified scenario, in particular the geodetic space and the mappings between infinite spaces and finite ones. Consider a parcel distribution company with warehouses distributed in Europe (for the sake of simplicity, we consider only six warehouses distributed in Italy and Spain). The company exploits vehicles to distribute parcels. Each vehicle is equipped with a device (mounting a GPS) that periodically notifies its position. Before reaching the final warehouse, vehicles can pass through intermediate warehouses. Each time a vehicle enters a warehouse, different operations have to be performed according to the country's rules, including the decision of the next warehouse the vehicle has to reach.

As depicted in Figure 7, the required spaces are: S_1, a geodetic spatial model covering the involved countries; S_2, a graph spatial model where each node corresponds to a specific warehouse (identifiers wh_1, wh_2, ..., wh_6) and each arc connects the warehouses that can be reached without any intermediate stop; finally, S_3, a name spatial model containing the identifiers of the two countries (Italy, Spain).

Explicit mappings are required from S_1 to S_2 with the aim of localizing each warehouse in the geodetic space. Mappings are in the form

$$M = \langle\langle S_1, \langle PREM, \langle [lat, long], radius, d\rangle\rangle\rangle, \langle S_2, \langle ENUM, \{wh_i\}, \in\rangle\rangle\rangle$$

where the target is an enumerative context containing a node of the S_2 space (i.e., a warehouse wh_i) and the source is a premetric declarative context specifying the area in S_1 where the warehouse wh_i is located. For example, the mapping $M_1 = \langle SC_1, SC_2\rangle$, where $SC_1 = \langle S_1, \langle PREM, \langle[45.523653, 9.219436], 50, d\rangle\rangle\rangle$ and $SC_2 = \langle S_2, \langle ENUM, \{wh_1\}, \in\rangle\rangle$. In Figure 7 spatial contexts on S_1 have been hugely enlarged for visualization purposes. Moreover, explicit mappings are required from S_2 to S_3 with the aim of localizing each warehouse in its country. Mappings are in the form

$$M = \langle\langle S_2, \langle ENUM, \{wh_i\}, \in\rangle\rangle, \langle S_3, \langle ENUM, \{country\}, \in\rangle\rangle\rangle$$

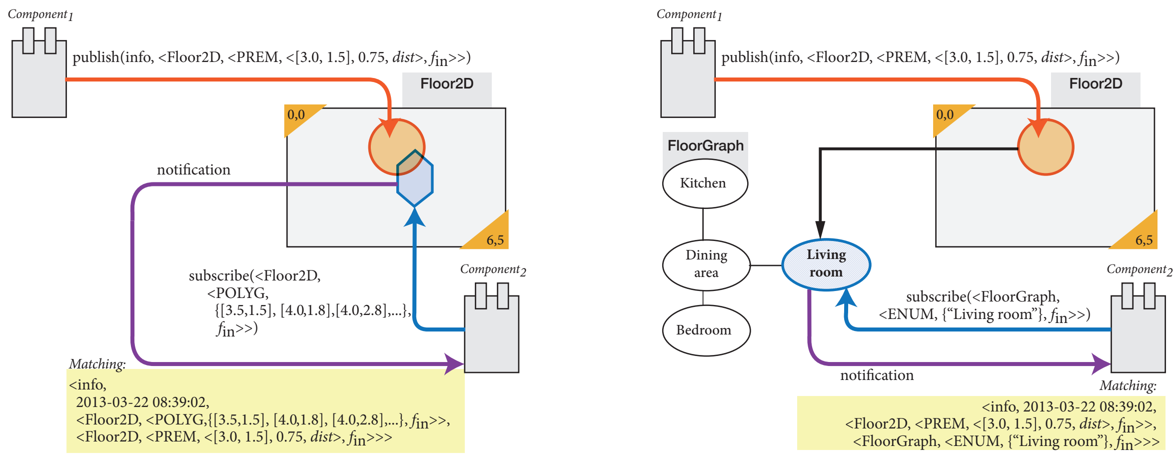
where the source is an enumerative context containing a node of the graph (i.e., the identifier of a warehouse) and the target is an enumerative context containing the identifier of the country. For example, the mapping $M_2 = \langle SC_2, SC_3\rangle$, where $SC_2 = \langle S_2, \langle ENUM, \{wh_1\}, \in\rangle\rangle$ and $SC_3 = \langle S_3, \langle ENUM, \{Italy\}, \in\rangle\rangle$. Finally, six indirect mappings are derived: their source contexts are the source contexts of the explicit mappings defined between S_1 and S_2, and their target contexts are the target contexts of the explicit mappings defined between S_2 and S_3. For example, $M_3 = \langle SC_1, SC_3\rangle$.

Two applications are required (App_1 for Italy and App_2 for Spain), each implementing the local rules. Each application subscribes to the appropriate country to be notified when a vehicle enters a warehouse in the country of competence. For example, App_1 performs a subscription to the enumerative context SC_S in S_3 containing the location Italy (i.e., $SC_S = \langle S_3, \langle ENUM, \{Italy\}, \in\rangle\rangle$). Periodically, the vehicles make publications in the S_1 space, thus sharing their position with all the interested applications. Publications are in the form

$$Pub = \langle vehicleID, \langle S_1, \langle PREM, \langle [lat, long], radius, d\rangle\rangle\rangle\rangle$$

where vehicleID is the thematic information that identifies the vehicle, and $\langle S_1, \langle PREM, \langle [lat, long], radius, d\rangle\rangle\rangle$ is a premetric declarative context specifying the position of the vehicle identified by vehicleID in the geodetic space. For example, publication $Pub_i = \langle 12345, SC_P\rangle$, where 12345 is the identifier of the vehicle that performs the publication and $SC_P = \langle S_1, \langle PREM, \langle[45.51788, 9.214071], 20, d\rangle\rangle\rangle$ is the location in which it has been localized.

When the vehicle 12345 makes the publication Pub_i, App_1 is notified because an indirect match occurs. Indeed, the following conditions result true: SC_P intersects SC_1, SC_1 is indirectly mapped to SC_3 (M_3 mapping), and SC_S intersects SC_3. When notified, App_1 receives the thematic information 12345 enriched with all the contexts that enabled the matching, that is, SC_P, SC_1, SC_2, SC_3, and SC_S. This way, App_1 is aware of the warehouse in which the vehicle 12345 is, and, if it is able to manage S_2, it can inspect the graph in order to



(a) Publish and subscribe in the same space

(b) Publish and subscribe in different spaces

Figure 6: Publication and subscription services

decide the next stop for the vehicle.

The finite-space version of SIS could not handle this application scenario for the presence of a geodetic space that is infinite. In any case, the operations used in this scenario are exactly the same as they would be used with the finite-space version to maintain backwards compatibility. What differs is how these operations are managed in the case involving infinite spaces. Obviously, using the finite-space version of SIS to handle this applicative scenario would have involved the use of a grid space to represent the geodetic space. The application would have had to manage all the correspondences between geodetic positions and locations on the grid space.

V. Implementation

This section will present issues related to the implementation of the current version of SIS, focusing on the new aspects that allow SIS to manage both finite and infinite spaces.

A. Layering

The developed prototype is based on the implementation of the finite-space version of Space Integration Services.

The infinite-space extension of SIS is organized according to two different software layers [8] on top of the finite-space SIS Core layer. Figure 8 shows the resulting structure.

The *Distributed Access* layer exhibits three different mechanisms allowing the interaction with the platform. The *Web Services* interface provides applications access to the platform features by means of the Representational State Transfer (REST) paradigm. The Web Services offers the SIS services through the `SISManager` class that implements the interfaces shown in the class diagram presented in Figure 9:

- `SpacesManagement` allows the creation, update, and deletion of spaces, as well as the management of the projections defined between spaces.

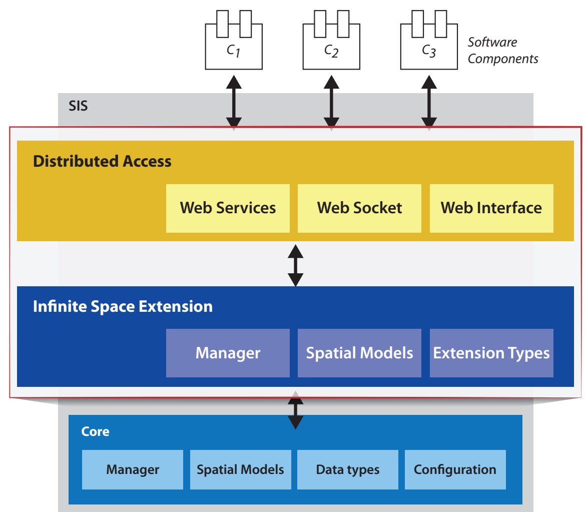


Figure 8: The extended SIS Structure

- `SpacesInspection` allows an application to explore the currently configured spaces, as well as the projections defined on those spaces.
- `MappingsManagement` provides an application with the necessary primitives for creating, editing, and removing mappings.
- `MappingsInspection` exposes the primitives for the exploration of the mappings (explicit and implicit) defined on the SIS instance; similarly to `SpacesInspection`, it offers a notification when a mapping is changed.
- `SpacesPublishSubscribe` exposes the primitives for the publication and subscription operations, as well as the necessary methods to enable the push notifications towards the applications.
- `MatchingNotifier` enables the asynchronous no-

tification (to applications) of new thematic information located in the subscribed contexts through the observer-observable pattern.

- `MappingChangeNotifier` enables the asynchronous notification (to applications) of mapping changes through the observer-observable pattern.

Web Sockets [9] are a recent W3C standard for two-way asynchronous communication in the context of web applications; this technology makes available asynchronous communication of notifications to the applications. Finally, the *Web Interface* visually exposes all the primitives of the platform and allows for the configuration of the users with their access permissions.

The *Infinite Space Extension* layer encloses the management of the new spatial models (*Spatial Models*), the new data structures (*Extension Types*), and the business logic to handle publications and subscriptions, the definition of contexts, and the creation of mappings between contexts of different spaces (*Manager*).

The *Core* layer contains the finite spatial models (i.e., graph, name, and grid) with related primitive types and the manager in charge of monitoring instances of SIS itself. In particular, the *Spatial Models* manages the finite spatial models, the *Data Types* manages the data structures, the *Manager* handles the subscription, publication, and the mappings, finally the *Configuration* handles the startup of the SIS instance. The *Core* layer uses the JESS rule engine in order to handle the operations of transitive closure and matching. This layer grants the full compatibility of the SIS extension with the previous versions of SIS.

The prototype has been developed in Java because the current Core layer is implemented in this language.

B. Inherited Basic Classes

The current developed prototype uses the base classes already defined in SIS. In particular the `Space`, `SpaceParameters`, and `Location` classes that serve as the basic foundation for describing the supported spatial models.

Every spatial model is strictly related to a `SpaceParameter` subtype, which defines the characteristic configuration parameters of that kind of space. As an example, a geodetic space will have the starting and ending coordinates defining the boundary of the region as parameters.

The `Space` class defines a set of common operations that could be performed on every kind of space instance; these operations include the calculation of the distance between two valid locations, the listing of the supported metrics, and the check of validity of a given location. The `Space` class also specifies that a space needs a name which will be used as the unique identifier.

C. Geodetic and Bi-dimensional Cartesian Spaces

In order to verify the suitability and applicability of the conceptual model described in Section II, two spatial models that are inherently infinite have been implemented: the Cartesian two-dimensional and the geodetic. The Cartesian space implementation provides as basic premetric the Euclidean metric and the related space parameter has been defined so that it specifies the boundary of the 2D region.

Two different models of the geodetic space have been provided according to two standard terrestrial representations defined by the European Petroleoum Survey Group (EPSG): the first called EPSG:3857 [10] (also known as Pseudo-Mercator) and the second called EPSG:4326 [11]. Both the representations use the WGS84 reference ellipsoid. The EPSG:3857 model is principally used to support tiled map representations of the World ([12], [13], [14]) as found in Google Maps, Bing Maps, OpenStreetMaps, and Nokia Here; whereas the EPSG:4326 model is used by the GPS satellite navigation system and for NATO military geodetic surveying.

The main difference between the two representations is that in EPSG:3857 the coordinate system is projective (i.e., the satellite coordinates are projected using the Mercator projection or the spherical projection Mercator), whereas EPSG:4326 is a reference geographic system [6] where the coordinates are not projected.

Both the geodetic spaces provide as basic premetric the orthodromic distance. The orthodromic distance is calculated using the haversine formula [15].

D. Infinite Space Management

Since the finite-space SIS Core has been used as the foundation of the prototypical implementation, some techniques have been required to manage infinite spaces in terms of finite spaces without incurring in the limitations of discretization. The use of a finite representation also enables a backward compatibility with all the already defined spatial models (i.e., name, grid, and graph spaces).

Techniques in the area of Geographic Information Systems (GIS) have been analyzed, in particular, the tessellation technique [16]. Tessellation is the process of tiling a plane using one or more geometric shapes (called tiles) with no overlaps and no gaps. Although the tessellation technique is well known and there exist implementations that overcome known problems of efficiency [17], this technique is difficult to apply since it is based on individual locations as the base unit of ”reasoning“ on spaces. Beside being in contrast with the model that considers the spatial context as the base unit, its exploitation will result in a huge memory occupancy because when defining mappings among spatial contexts, all the locations involved will be stored.

The second analyzed technique is partially inspired by [18] and [19] and overcomes the problems that tessellation introduces. The proposed technique is based on the following considerations that directly derive from the conceptual model:

- The base unit used is the spatial context; each operation the platform supports is based on spatial contexts; each operation on a space, either finite or infinite, can be reduced to a sequence of operations on spatial contexts.
- A spatial context may contain one location only.
- The number of effective spatial contexts in a space is finite.

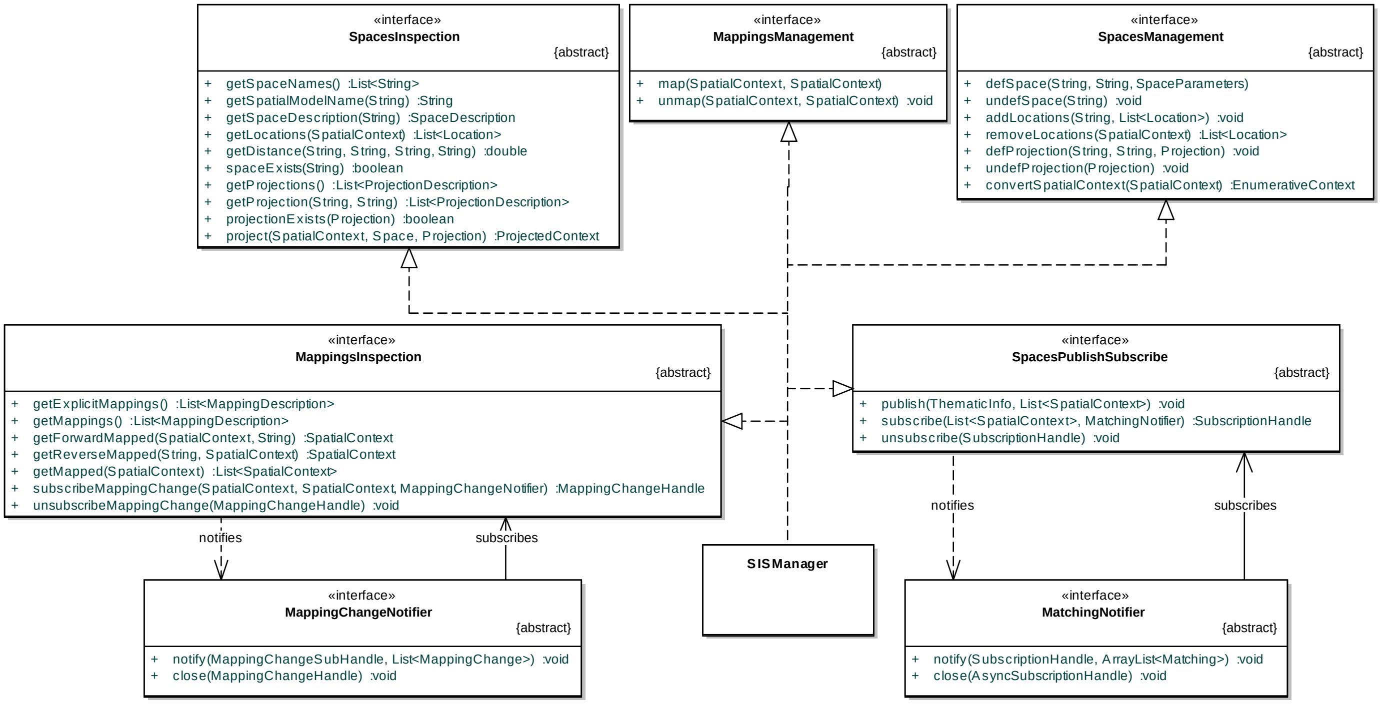


Figure 9: SIS services

- Each spatial context can have a unique identifier; if two contexts share the same identifier then they describe the same region of space.

Given the above considerations, the key idea is to use a finite space (in particular a name space) in the finite-space SIS Core to represent an infinite space. The proposed approach makes use of two concepts: (1) *spatial context fingerprint* (SCF) and (2) *finite support space*. The spatial context fingerprint is a unique identifier of a spatial context that is built from its characteristic locations and membership function. In particular, current implementation uses hash functions [20] to generate the identifier. Each SCF is maintained as a location of a space built form a name spatial model and termed finite support space. Each infinite space has its own finite support space containing the SCFs of all the defined contexts in that space. Obviously, a relation between the original spatial context and the fingerprint inserted in the support space must be maintained.

As an example consider Figure 10. Suppose that the application needs to create the spaces $\mathtt{S}_1$ and $\mathtt{S}_2$ as specified in the application scenario presented in Section IV. They are respectively a geodetic and a graph space. To define the spaces, the application relies on the `defSpace` primitive as follows:

```
defSpace(S1, GeodeticSpatialModel,
    Boundary.WORLD)

defSpace(S2, GraphSpatialModel,
    <{wh1, wh2, wh3, ...},
     {wh1-wh2, wh2-wh3, ...}>)
```

In order to represent the geodetic space $\mathtt{S}_1$ in the finite SIS Core, a new space ($\mathtt{support\_S}_1$) based on the name spatial model is created. For each spatial context defined in $\mathtt{S}_1$, a corresponding SCF is defined as location in the support space $\mathtt{support\_S}_1$. In the example, the $\mathtt{SC}_P$ context defined in $\mathtt{S}_1$ will be represented by the location `SC_E7984AD89...` in the $\mathtt{support\_S}_1$ space. On the opposite, the graph space is directly created in the finite SIS Core.

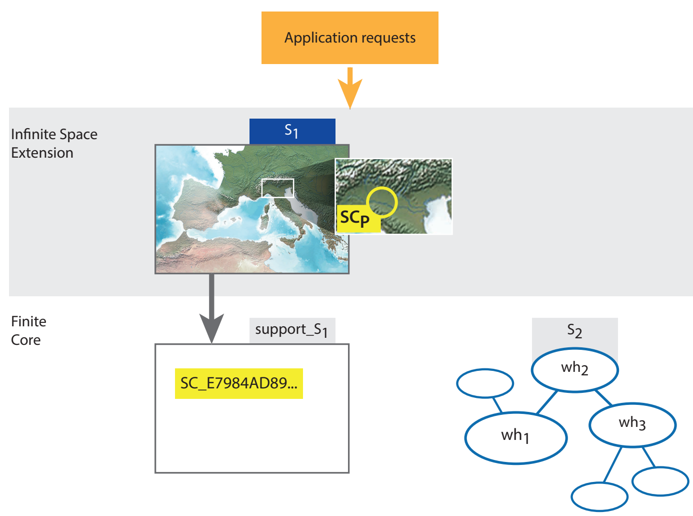


Figure 10: Definition of infinite spaces: an example

In conclusion, an infinite space is managed through a finite space representing the collection of the defined spatial contexts. This approach maintains compatibility with all the originally supported finite spaces and does not impact on the publish and subscribe mechanism offered by the SIS Core.

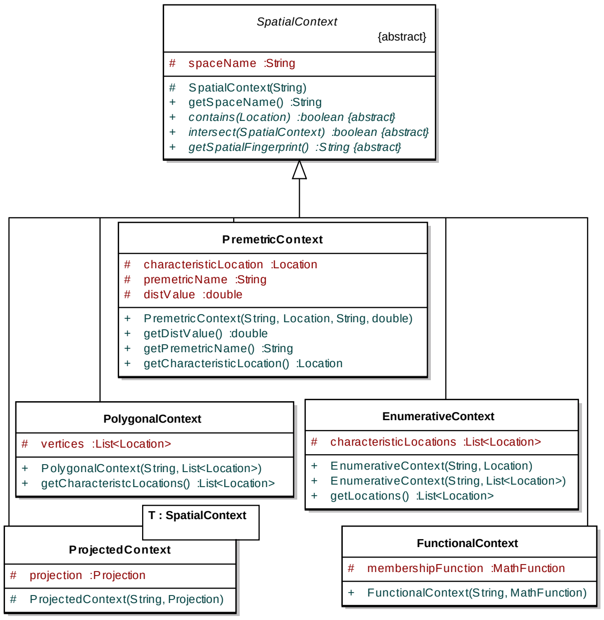


Figure 11: Hierarchy of spatial contexts

E. Spatial Contexts and Projections

As explained in the Section II, each spatial context is identified by a set of characteristic locations and a membership function. Accordingly, the spatial contexts supported by the current implementation are those shown in Figure 11.

An enumerative context (`EnumerativeContext` class) is just a list of locations, whereas a premetric context (`PremetricContext` class) is specified by means of a single location and a radius. Both enumerative and premetric contexts can be used on infinite and finite spaces.

In the prototypical implementation, the polygonal (`PolygonalContext` class), functional (`FunctionalContext` class), and projected (`ProjectedContext` class) contexts are limited to infinite spaces. The polygonal context represents a polygonal closed region of the space. It is defined by providing the vertexes of the region from which the boundaries are linearly interpolated. Vertexes should be provided so that the resulting edges do not overlap. The intersection test between polygonal contexts is done using state of the art techniques in the field of Computational Geometry. In the proposed prototype the general polygon clipper technique [21] has been used.

The functional context represents a region of the space using a mathematical function, which currently is limited to a two variable inequality. This function is internally represented by the `MathFunction` class, which makes use of the exp4j library [22] to enable the evaluation of the mathematical function. The evaluation is done with the Dijkstra version of the shunting yard algorithm ([23], [24]). The usage of this approach enables the specification of the mathematical function using a string representation that is also suitable for the web service interaction.

Finally, the projected context derives from the application of a projection function on a source spatial context. Projections (that are instances of the `Projection` class) are specified by a mathematical function (that operates on locations) and by a source and a target spatial model. The mathematical function is realized by means of the `MathFunction` class. A projection may specify the same spatial model both for the source and the target. It is useful, for example, to define a projection that reifies a rotation defined on a Cartesian space or, more trivially, to define a projection that realizes the identity operation on a space. The characteristic locations of a projected context are those obtained by applying the projective function of the `Projection` to all the characteristic locations of the source spatial context. The membership function for a location in a projected context relies on the membership function for the corresponding (i.e., inverse projected) location in the source space, thus the invertibility constraint on the projection function. To obtain a `ProjectiveContext`, an application can rely on the `project` primitive defined in the `SpacesInspection` interface (see Figure 9).

Finally, to support mappings, subscriptions, and publications, the three following methods have been defined in the base class `SpatialContext`:

- `contains`, which checks if a specified location in the space is included in the context.
- `intersect`, which enables the matching operation on contexts defined on infinite spaces.
- `getSpatialFingerprint`, which returns the spatial context fingerprint (SCF).

F. Mapping Management

As in the finite-space case, the definition of a mapping enables the creation of one or more relations between regions of different spaces. The creation of a mapping involving infinite spaces includes the following two steps:

1) The spatial context fingerprint is computed for each context and the corresponding locations are added to the corresponding finite support spaces (unless they have already been defined).
2) A finite-mapping request is sent to the finite-space SIS Core for the actual creation of the mapping.

This solution allows both the new spatial models and the already existing finite spatial models to be deal with in a similar way, so that both types can be used as source and target contexts for the mapping.

In order to clarify on how the mappings are handled, the mapping M_1 defined in the application scenario presented in Section IV will be deeply discussed. To define the mapping, the application relies on the `defMapping` primitive as follows:

defMapping(SC_1, SC_2)

where

SC_1 =
<S_1, <PREM, <[45.523653, 9.219436], 50, d>>>
SC_2 =
<S_2, <ENUM, {wh_1}, $\in$>>

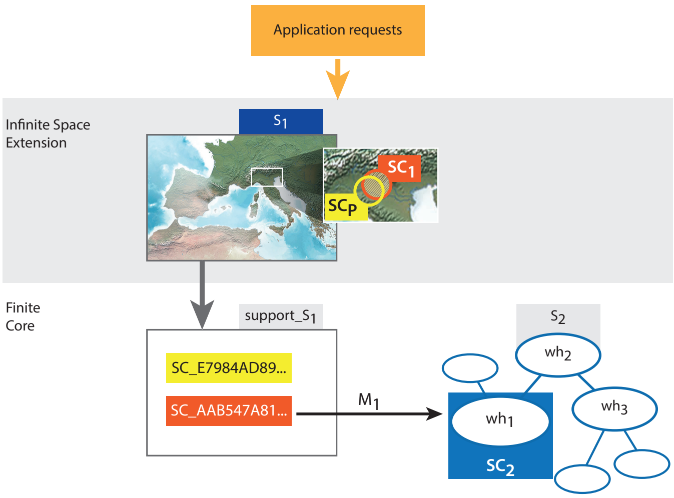


Figure 12: Definition of mappings: an example

As sketched in Figure 12 (that enriches Figure 10), the infinite extension defines a new location `SC_AAB547A81...` in the support space `support_S`$_1$ for the spatial context `SC`$_1$ (that is defined on the infinite space `S`$_1$). Successively, the mapping `M`$_1$ is created between the contexts `<support_S`$_1$`, <ENUM, SC_AAB547A81..., ∈>>` and `<S`$_2$`, <ENUM, wh`$_1$`, ∈>>`.

G. Publication

A generic publication on an infinite space (in the following, publication space) is converted into a publication on the corresponding finite support space as follows:

1) An intersection test is performed between the publication context and the source context of each mapping defined in the publication space. Only the source contexts are explicitly considered thanks to the fact that the finite-space SIS Core already contains all the implicit mappings. The result is a set (possibly empty) of intersecting contexts.
2) A new publication context is defined with all the fingerprints of the intersected contexts. The context is an enumerative one defined in the support finite space of the original publication space. Such a new context will be used to define a new publication whose thematic info contains the original thematic info plus the original infinite publication context so that, when a notification will be performed, the original context can be restored.
3) The newly created publication is delivered to the finite-space SIS Core to the reasoning process.

The above operations are performed for each spatial context of the publication that has been defined in an infinite space.

H. Subscription and Notification to Applications

A subscription is made on one or more spatial contexts. It involves the creation of a communication channel for receiving the notifications (for example, a communication channel could be created using Web Sockets). Each subscription is stored until the application removes it.

A subscription request is handled in a way similar to the publication case. For each spatial context defined in a infinite space (in the following, subscription space), the following operations are performed:

1) An intersection test is performed between the subscription context and the target context of each mapping defined in the subscription space. The result is a set (possibly empty) of intersecting contexts.
2) A new subscription context is defined with all the fingerprints of the intersected contexts. The context is an enumerative one defined in the support finite space of the original subscription space.
3) The newly created subscription is delivered to the finite-space SIS Core to be managed.

The deletion of a subscription closes the communication channel and removes all the support subscriptions created.

The notification process is performed only if the spatial contexts involved in a publication are in direct matching with the subscription contexts already stored. Only the direct matchings are considered because the finite-space SIS Core calculates the indirect matching in the transitive closure.

The construction of a notification is a bit more complicated because the contexts reported by the matching process can contain both finite and infinite spaces. Since the core implementation internally operates only on finite spaces, a check on the defined infinite spaces needs to be done in order to convert the involved contexts in the finite support spaces into the original contexts on infinite spaces.

The infinite-space layer of the SIS platform acts as the receiver for all the notifications raised by the finite-space SIS Core. Each notification includes the list of the corresponding matchings. A matching contains the published thematic information and all the contextualization information (i.e., all the contexts that have been found during the matching operation). Each notification is then checked to find possible contexts defined in finite support spaces. Each of these contexts, identified by its fingerprint, is then substituted with the original finite one. The resulting notification is then made available to the subscribed applications through the communication channel.

VI. Performance Evaluation

Several performance tests have been conducted aimed at comparing the mean reasoning times registered by the SIS prototype (SIS 2 in the following) and by the previous version based on finite spaces only (SIS 1 in the following). The mean reasoning time is the mean time between the reception of a publication and the moment at which notifications are made available to interested applications. The mean reasoning time is evaluated with respect to both the number of mappings and the size of the context. Since the contexts involved in the tests are defined over infinite spaces, they have been approximated in SIS 1 as it will be discussed in the following subsections describing the setup of the experiments and the achieved results.

The experimental tests have been performed on an SIS 1 and a SIS 2 instances running on a desktop PC equipped

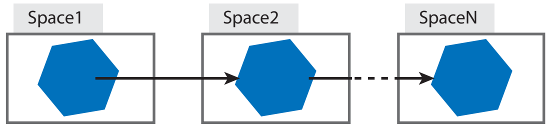


Figure 13: Space configuration for the mapping test

with an Intel Core i5 2.8 GHz, 4GB of RAM, Windows(R) 7 64bit, and the 64bit version of the Java Runtime Environment 1.6.33. Publications were generated by a client and sent to the SIS server. Moreover, performance tests have been also conducted on an Intel(R) Next of Computing Unit (NUC) equipped with an Intel(R) Core i3 1.8 GHz processor, 8GB of RAM, Windows(R) 8.1 64bit, and the 64bit version of the Java Runtime Environment 1.7.0 build 45. Indeed, such an hardware platform can be easily integrated in an actual pervasive computing environment because of its characteristics that make it a good compromise between performance and dimensions.

A. *Mean Reasoning Time vs. Mappings*

The first experimental setup allows studying the dependence of the mean reasoning time on the number of mappings. In this test, the mappings are created in the configuration phase and do not change dynamically. In particular, we analyzed the mean reasoning time in two different configurations: the mappings between polygonal contexts, and mappings between premetric declarative contexts.

In both the configurations, SIS 2 includes n Cartesian spaces, each containing a single context. Every context is directly mapped onto a context in the next space, thus realizing a chain of $(n-1)$ explicit mappings, as shown in Figure 13 for the polygonal contexts. On SIS 1 the Cartesian space has been approximated using a grid space and each context is approximated with a number of cells equals to the area occupied by context itself.

Considering the implicit mappings, the total number of mappings is therefore equal to $n(n-1)/2$. Publications occur on the context in the first space, whereas the subscription is made on the context in the last space. With this generic configuration, the mean reasoning time for a publication can be measured as a function of the number of spaces n. In the experimental tests, n varies from 2 to 150. Obviously, in actual applications, a chain of 150 mappings is to be considered an exceptional case: the objective of this configuration was to stress the system only.

In the first configuration, each Cartesian space in SIS 2 contains a single hexagonal polygonal context inscribed in a circle with a radius of two units. On SIS 1, each context is approximated with a number of cells equals to the area occupied by the polygonal context (i.e., 16 cells).

Figure 14 and Figure 15 show the mean reasoning time (expressed in milliseconds), using polygonal spatial contexts, registered by both the implementations (SIS 1 and SIS 2) when running on the desktop PC and the NUC respectively. As expected, the mean reasoning time obtained from SIS 2 deployed on the desktop PC is lower than that obtained when

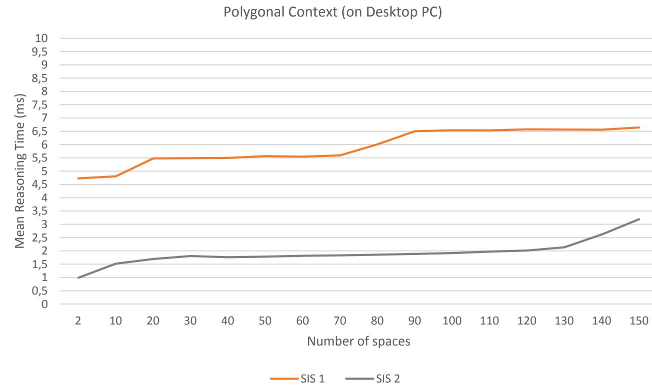


Figure 14: Mean reasoning time vs number of spaces on the desktop PC - Polygonal context

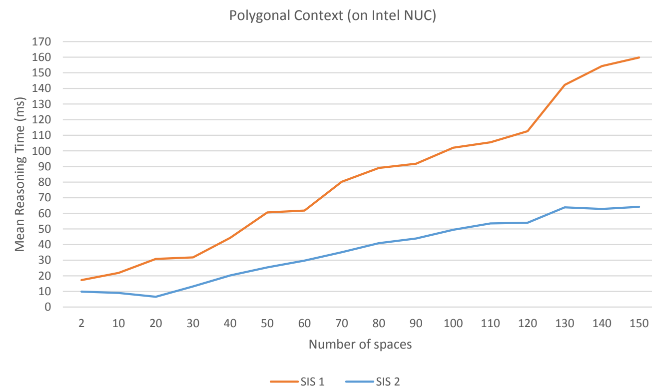


Figure 15: Mean reasoning time vs number of spaces on the Intel NUC - Polygonal context

deployed on the Intel NUC: an average of 5.91 ms for SIS 2 on desktop PC vs an average of 81.63 ms for SIS 2 on Intel NUC. Moreover, the mean reasoning time on the desktop PC is almost constant during the execution of the test, whereas on the NUC is almost sublinear. Finally, as pointed out by both the figures 14 and 15, SIS 2 is faster that SIS 1 using a grid approximation. In particular, SIS 2 on desktop PC is 3.99 ms faster than SIS 1, whereas SIS 2 on Intel NUC is 45.27 ms faster than SIS 1.

In the second configuration, each Cartesian space in SIS 2 contains a single premetric declarative context with a radius of two units. On SIS 1, each context is approximated with a number of cells equals to the area occupied by the premetric declarative context (i.e., 16 cells).

Figure 16 and Figure 17 show the mean reasoning time (expressed in milliseconds), using premetric spatial contexts, registered by both the implementations (SIS 1 and SIS 2) when running on the desktop PC and the NUC respectively. As in the previous configuration, the mean reasoning time obtained from SIS 2 deployed on the desktop PC is lower than that obtained when deployed on the Intel NUC: an average of 5.91 ms for SIS 2 on desktop PC vs an average of 81.63 ms for SIS 2 on Intel NUC. Again, the mean reasoning time on the

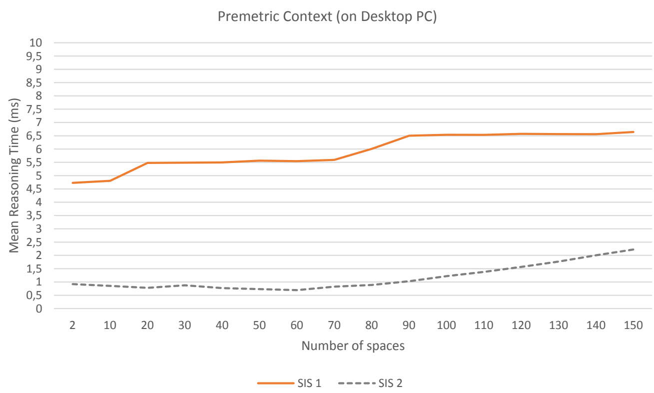


Figure 16: Mean reasoning time vs number of spaces on the desktop PC - Premetric context

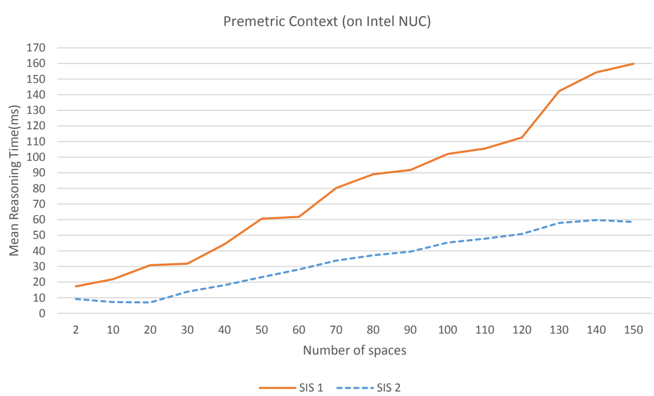


Figure 17: Mean reasoning time vs number of spaces on the Intel NUC - Premetric context

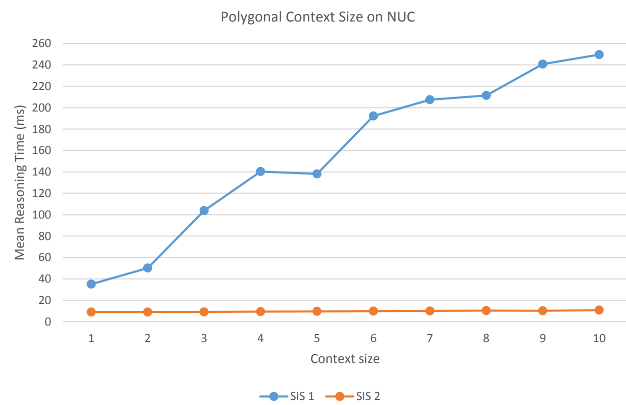


Figure 18: Mean reasoning time vs. size of context on the Intel NUC

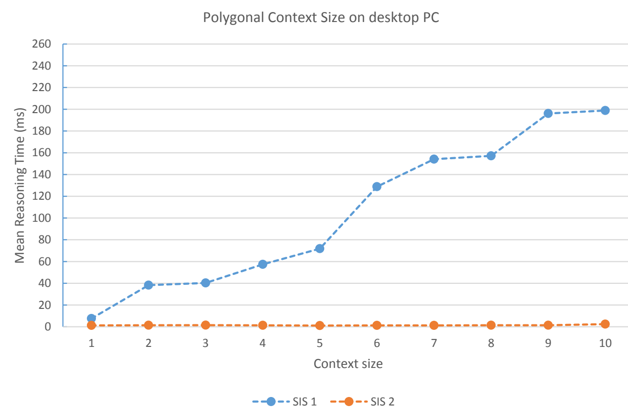


Figure 19: Mean reasoning time vs. size of context on the Desktop PC

desktop PC is almost constant during the execution of the test, whereas on the NUC is almost sublinear. The mean reasoning time for premetric contexts and for polygonal contexts are equal because the underlying implementation is the same as discussed in Section V: it consists in a single spatial context fingerprint for each defined spatial context. Finally, as in the previous configuration, SIS 2 is faster that SIS 1 using a grid approximation. In particular, SIS 2 on desktop PC is 4.07 ms faster than SIS 1, whereas SIS 2 on Intel NUC is 46.06 ms faster than SIS 1.

B. Mean Reasoning Time vs. Context Size

The second test aims to investigate the dependence of the mean reasoning time on the size of the contexts.

The space configuration in this case includes two Cartesian spaces for SIS 2 and two bi-dimensional grids for SIS 1. One hexagonal polygonal context was defined in each space, the first one for publication and the second one for subscription. As in the previous test, the hexagonal polygon has been approximated with a number of cells equals to its area on the grids. During the test, the number of mapped spaces has been maintained fixed at two, whereas the radius of the circle used to calculate the vertices of the inscribed polygonal contexts was varied according to the values between 1 and 10.

As pointed out by Figure 18 and Figure 19, the SIS 2 prototype has a constant mean reasoning time; about 2 ms on the desktop PC and around 9ms on the Intel NUC, while the SIS 1 implementation behaves like $O(24n)$ on the Intel NUC and $O(20n)$ on the desktop pc.

As expected, the SIS 2 prototype presents a constant trend independent from the context size because every context is represented using a single entity.

C. Discussion

The experimental tests show how the implemented techniques allow for a better management of different context sizes in infinite spaces, maintaining a constant time of reasoning. Moreover, the prototype has a better scaling behavior with respect to the finite-space implementation, both for increasing number of mappings and for increasing size of publication and subscription contexts. Such improved scalability can be understood in terms of the number of facts that form the knowledge base of the rule engine managing the operations of transitive closure and matching; the infinite-space implementation reduces the number of facts to one per context, and thus to two facts for a single mapping, whereas the finite-space implementation creates a fact for each location involved in the mapping.

Finally, the execution of the first test of SIS 2 (number of mappings vs mean reasoning time) states that the Desktop PC has been faster than the NUC regardless of the type of context. This is obviously true, but it is reasonable to assume that in actual situations, the chain of mappings will contain a number of mappings that is substantially lower than 150. By considering a chain of 10 mappings (a plausible value), the mean reasoning time is approximately equal to 9 ms (about 3,5 times compared to the PC). Although each application has its own time constants, we consider a delay of 9 ms to be acceptable for pervasive computing applications.

VII. Related work

Location-aware computing has been an active area of research. Different platforms at the state of art enable location-aware applications focusing on sensor fusion and reasoning with the help of a multi-spatial model, or hybrid model (as called by Becker et al. [25]).

Location Stack [26] defines a layered model for fusing location information from multiple sensors and reasoning about an object's location. It, however, does not incorporate a spatial model of the physical world and does not support representations of immobile objects. This leads to a lack of support for spatial reasoning relative to stationary entities such as rooms or corridors.

Loc8 [27], on the other hand, extends the Location Stack layered architecture by considering only high level position and data instead of low level sensor data. Reasoning is applied to that position data, enriched by the knowledge given by a base ontology, to infer additional spatial relationships.

The *Aura Space Service* [28] combines coordinate and hierarchical location models into a single hybrid model, which supports spatial queries. The focus of the Aura Space Service is only on modeling the physical space and supporting spatial queries. It does not address location inferencing and does not provide a framework for spatial reasoning.

MiddleWhere [29] uses the hybrid location model introduced by the Aura Space Service and enables the fusion of different location sensing technologies. MiddleWhere introduces also probabilistic reasoning techniques to resolve conflicts and deduce the location of people given different sensor data. The model of the world is stored in a spatial-enabled database.

Semantic Spaces from Microsoft Research decomposes the physical environment into a hierarchy of spaces. The locations of moving users or devices are correlated to actual physical spaces, thus it is capable of answering "containment" queries. However, because of its inherent lack of metric attributes and precision, it is unable to compute distance accurately or represent locations precisely, which are requirements for some ubiquitous computing applications [30].

Semantic Spaces and Location Stack lack any support for infinite spaces and in general spaces with a coordinate system, while Loc8 and MiddleWhere have at least one spatial model with a Cartesian coordinate system and can handle different levels of precision on that space model. These two platforms substantially treat infinite spaces by using different granularities for location representation on a local and a global coordinate system.

VIII. Conclusion and Future Work

The paper proposed an extension to the conceptual model of the SIS platform. This refinement comes in order to enable the use of infinite spatial models like the geodetic or the Cartesian ones.

The approach that has been presented involves the extension of the concept of spatial context and the use of that concept as the elementary unit at the basis of all the spatial operations enabled by the platform itself (i.e., mapping and matching).

With the help of a prototypal implementation the revised conceptual model has been tested for performance evaluation. The tests that have been conducted showed an overall increase of performance and capacity to handle spatial contexts with large extent as needed when using the geodetic space. Moreover, the tests have also highlighted that the actual prototypal implementation can be installed into a computing machine (e.g., an Intel NUC) that can be easily plunged in an environment: the mean reasoning time with respect to both the number of mapped spaces and the dimension of the contexts can be considered negligible with respect to the timing requirements of pervasive computing applications.

The main future work consists in the deep integration of the Infinite Space Extension layer in the Core layer. This will enable a more efficient use of the rule engine. Moreover, the platform will be exploited to provide spatial localization to properties related to domain entities as inferred from stimuli acquired by sensing devices. This results in a framework that will support the implementation of pervasive computing applications. Preliminary results can be found in [31].

References

[1] D. Bernini, F. Fiamberti, D. Micucci, F. Tisato, and A. Vertemati, "Spaces-based communication: an extension to support infinite spatial models," in UBICOMM 2013: The Seventh International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2013, pp. 92–97.

[2] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," Computer, vol. 36, no. 3, 2003, pp. 25–31.

[3] L. Bullivant, Responsive environments: architecture, art and design. Victoria & Albert Museum, 2006.

[4] D. Bernini, F. Fiamberti, D. Micucci, and F. Tisato, "Architectural abstractions for spaces-based communication in smart environments," Journal of Ambient Intelligence and Smart Environments, vol. 4, no. 3, 2012, pp. 253–277.

[5] C. A. Patterson, R. R. Muntz, and C. M. Pancake, "Challenges in location-aware computing," Pervasive Computing, IEEE, vol. 2, no. 2, 2003, pp. 80–89.

[6] Spatial referencing by geographic identifiers, ISO Standard ISO 19 112, 2003, accessed: 30.5.2014. [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26017

[7] C. Cauvin, "A systemic approach to transport accessibility. A methodology developed in Strasbourg: 1982-2002," Cybergeo: European Journal of Geography, Systems, Modelling, Geostatistics, no. 311, 2005.

[8] D. Garlan and M. Shaw, "An Introduction to Software Architecture," Advances in Software Engineering and Knowledge Engineering, vol. 1, 1993, pp. 1–40.

[9] I. Fette and A. Melnikov. The WebSocket Protocol - RFC 6455. Accessed: 30.5.2014. [Online]. Available: http://datatracker.ietf.org/doc/rfc6455/?include_text=1 (2011)

[10] H. Butler, C. Schmidt, D. Springmeyer, and J. Livni. Epsg geodetic parameters: 6864. Accessed: 30.5.2014. [Online]. Available: http://spatialreference.org/ref/sr-org/6864/ (2013)

[11] ——. Epsg geodetic parameters: 4326. Accessed: 30.5.2014. [Online]. Available: http://spatialreference.org/ref/epsg/4326/ (2013)

[12] A. Aitchison. The Google Maps/Bing Maps Spherical Mercator Projection. Accessed: 30.5.2014. [Online]. Available: http://alastaira.wordpress.com/2011/01/23/the-google-maps-bing-maps-spherical-mercator-projection/ (2011)

[13] OpenLayers. Spherical Mercator. Accessed: 30.5.2014. [Online]. Available: http://docs.openlayers.org/library/spherical_mercator.html/ (2008)

[14] P. P. Klokan. Tiles la Google Maps: Coordinates, Tile Bounds and Projection. Accessed: 30.5.2014. [Online]. Available: http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/ (2008)

[15] D. F. R. Sinnott, R.W; Olson, "Virtues of the haversine," Sky and Telescope, vol. 68, no. 2, 1984, p. 159.

[16] R. Laurini and D. Thompson, Fundamentals of spatial information systems, ser. The A.P.I.C. series. London, San Diego, New York: Academic press, 1992.

[17] K.-T. Chang, Introduction to geographic information systems (4. ed.). McGraw-Hill, 2008.

[18] H. Alt and L. Guibas, "Discrete Geometric Shapes: Matching, Interpolation, and Approximation," in Handbook of Computational Geometry. Amsterdam: Elsevier Science Publishers B.V. North-Holland, 1999, pp. 121–153.

[19] M. T. Goodrich and K. Ramaiyer, "Geometric Data Structures," in Handbook of Computational Geometry. Amsterdam: Elsevier Science Publishers B.V. North-Holland, 1999, pp. 463–489.

[20] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen, Introduction to algorithms. The MIT press, 2001.

[21] A. Murta. A general polygon clipping library. Accessed: 30.5.2014. [Online]. Available: http://www.cs.man.ac.uk/~toby/alan/software/ (2000)

[22] F. Asseg, "exp4j: a simple mathematical expression parser for java," 2012, accessed: 30.5.2014. [Online]. Available: http://www.objecthunter.net/exp4j/index.html

[23] E. W. Dijkstra, Primer of Algol 60 programming. Academic Press, Inc., 1962.

[24] R. R. Redziejowski, "On arithmetic expressions and trees," Communications of the ACM, vol. 12, no. 2, 1969, pp. 81–84.

[25] C. Becker and F. Dürr, "On location models for ubiquitous computing," Personal and Ubiquitous Computing, vol. 9, no. 1, 2005, pp. 20–31.

[26] D. Graumann, W. Lara, J. Hightower, and G. Borriello, "Real-world implementation of the location stack: The universal location framework," in Proceedings of the Fifth IEEE Workshop on Mobile Computing Systems and Applications, 2003, pp. 122–128.

[27] G. Stevenson, J. Ye, S. Dobson, and P. Nixon, "Loc8: a location model and extensible framework for programming with location," Pervasive Computing, IEEE, vol. 9, no. 1, 2010, pp. 28–37.

[28] C. Jiang and P. Steenkiste, "A hybrid location model with a computable location identifier for ubiquitous computing," in Proceedings of the 4th International Conference on Ubiquitous Computing, ser. UbiComp '02. London, UK, UK: Springer-Verlag, 2002, pp. 246–263.

[29] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. Mickunas, "Middlewhere: a middleware for location awareness in ubiquitous computing applications," in Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware. Springer-Verlag New York, Inc., 2004, pp. 397–416.

[30] B. Brumitt and S. Shafer, "Topological world modeling using semantic spaces," in Proceedings of the Workshop on Location Modeling for Ubiquitous Computing, UbiComp, 2001, pp. 55–62.

[31] F. Fiamberti, D. Micucci, M. Mobilio, and F. Tisato, "A layered architecture based on previsional mechanisms," in ICSOFT: International Joint conference on Software Technologies, 2013, pp. 354–359.

Analysis of Relation between Psychological Stress and Intentional Facial Expressions Based on Bayesian Networks

Kazuhito Sato, Hiroaki Otsu, and Hirokazu Madokoro
Department of Machine Intelligence and Systems Engineering,
Faculty of Systems Science and Technology, Akita Prefectural University
Yurihonjo, Japan
{ksato, m13a006, madokoro}@akita-pu.ac.jp

Sakura Kadowaki
Smart Design Corp.

Akita City, Japan
sakura@smart-d.jp

Abstract **- This paper presents a gender-specific stress model to analyze psychological stress factors affecting intentional facial expressions. We extract three facial expressions (i.e., happiness, anger, and sadness) from the basic six facial expressions. Then we present a graphical model of the relation between these three facial expressions and the psychological stress factors. From probabilistic reasoning based on the observed values of each facial expression, the following trends were obtained. The stress models constructed by Bayesian networks showed trends of different stress factors between men and women in relation to expression levels and psychological stress. The stress models appeared on happy faces of "lassitude" factor in men. The angry faces of "displeasure and anger" were affected by stress factors in women. Furthermore, the influence of "displeasure and anger" readily appeared in the upper part of the face when expressing "anger". In addition, the influence of "lassitude" strongly appeared in the lower part of the face with the expression of "sadness", in the upper part of the face with the expression of "happiness".**

Keywords - Bayesian networks; Self-organizing maps; Adaptive resonance theory; Expression levels; SRS-18; Intentional facial expressions

I. INTRODUCTION

Modern human life has become increasingly convenient along with the rapid development of our advanced information society. However, we are sometimes confused amid the plethora of information spreading around us. Products and artificial environments that we do not want to use are useless, even if they might be convenient. Using unattractive products makes us feel uncomfortable and stressed. Anyone might feel stress if forced to use unsatisfactory products.

Modern society is full of stress. Numerous people live their lives with various stressors: factors that cause stress. Stress is a biological reaction that develops when one confronts a psychological or spiritual stressor. Reactive processes of people interacting with the environment signify individual cognitive processes that are involved in biological reactions and physiological processes [1]. Figure 1 presents three indices used for human measurements in the areas of biology, psychology, and action. All require the consideration of individual processes when a person is subjected to stress. Moreover, we can posit a psychological index that shows stress to be psychological. As Figure 1 shows, various somatic symptoms appear with gastralgia and breathless states when stress accumulates. Stress appears from a psychological state when one experiences frustration and unease. In addition, stress is manifested in actions such as the increased consumption of alcohol and cigarettes. Stress affects the brain. Usually, the human brain responds effectually to maintain mental and physical balance. However, excessive stress can trigger mental illness such as depression [2]. Because it can be difficult to conceal stress, faces are often described as a window by which one can discern information of various types such as the state of a person's mind and health condition. Especially, facial expressions can reveal aspects of internal psychology, reflective emotions such as delight, anger, sorrow, pleasure, and the existence of stress. Close friends and family members communicate while interpreting stress from conditions and changes of facial expressions.

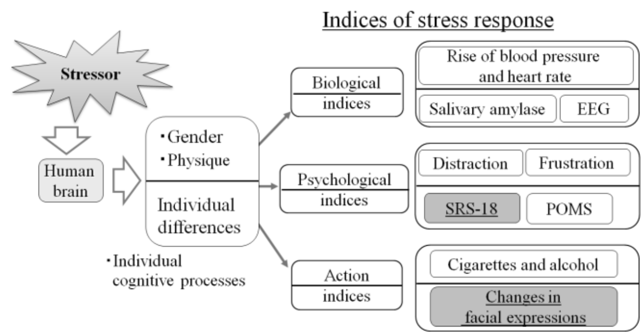


Figure 1. Background.

For this study, we specifically examine intentional facial expressions. Moreover, we set the upper part, lower part, and whole parts of the face as regions of interest (ROIs) to address static and dynamic diversity, as defined by Akamatsu [3]. We quantify the relation between facial expressions and psychological stress using Bayesian networks (BNs), which can describe stochastic relations of events as a graphic structure. For our evaluation experiment, we create stress models by gender. Then we analyze stress factors and stress elements in facial expressions of various types: happiness, anger, and sadness.

This paper is presented as the following. We review related work to clarify the position of this study in Section II. In Section III, we explain relations of psychological stress and facial expressions by FESCs that quantitatively represent individual facial expression spaces proposed in a previous paper. In Section IV, we explain our originally developed facial expression datasets including stress measurements and describe the method to capture facial expression images, preprocessing, classification of facial expression patterns with self-organizing maps, integration of facial expression categories with fuzzy adaptive theory. We describe how to create a model of stress elements using Bayesian Networks in Section V. We show analytical results of stress elements models for men and women in Section VI. In Section VII, we estimate stress factors using a male and female model with emphasis on facial parts effect of facial expressions. Finally, we present conclusions and intentions for future work in Section VIII.

II. RELATED STUDIES

Existing methods for measuring stress are divisible into two types: contact type and noncontact type. Using findings demonstrating that stress can affect amylase secretion in saliva, methods for quantifying stress have been proposed. Many products have been marketed based on such findings [4] [5]. In association with autonomic nerve activity and stress, many analytical studies of heart rate variability have been conducted. In fact, many devices have been developed and used in research and clinical practice [6] [7]. Numerous reports describe the correlation between stress and neural activity of the cerebrum. Particularly, electroencephalography (EEG) is a method that has been studied for many years. Studies using EEG have been performed in the context of the psychological state, such as attention and concentration, stress, and anxiety [8] [9]. As a new functional brain analysis method, near-infrared spectroscopy (NIRS), used for measuring cerebral blood volume changes locally and non-invasively, is also attracting attention as a stress measurement method, which indicates that the activity of the prefrontal cortex has changed significantly [10]. However, the methods for objective comparisons that are useful to identify the most suitable approach for stress measurements is extremely complex because each researcher has a different perspective: the type of stressor used in their experiments differs for each approach. In addition, methods using physical contact generally require a hardware interface.

Stress measurement checking using a questionnaire form is a popular noncontact measurement method. The Profile of Mood States (POMS) [11] is an inventory that is recognized worldwide, although its results cannot generally be compared because the target attributes differ. POMS consists of 65 items. The brief version of POMS consists of 30 items [12]. Actually, POMS becomes extremely burdensome if one incorporates images of facial expressions. Suzuki et al. developed the Stress Response Scale-18 (SRS-18) [13], which is useful for widely various subject ages. SRS-18 comprises 18 items and measures psychological stress encountered over a short time. Moreover, SRS-18 shows highly discriminative capability in high stress and low stress groups. Although most existing stress evaluations are aimed at the assessment of clinical conditions, numerous questions of the SRS-18 are related to events that normal people encounter on a daily basis.

Facial expressions and their intensities differ according to mental state, context, situation, etc. We actively use those differences to take facial expression images of each subject over a long term. We are aiming at describing and quantifying the relations of psychological stress and intentional facial expressions from temporal changes of long-term facial expression datasets. The degrees of expression levels differ among people and their feelings at any particular time. Therefore, the SRS-18 appears to be an optimal and helpful inventory for use in assessing facial expressions because of the measurement of physical and mental reactions and smaller number of question items.

III. RELATIONS OF PSYCHOLOGICAL STRESS AND FACIAL EXPRESSIONS

An appropriate amount of stress improves abilities of concentration and engenders work efficiency. However, excessive stress produces psychosomatic abnormalities because of humans' limited adaptive capability. The degree to which one feels stress effects reportedly varies subtly in similar environments because of individual differences from conditions and tolerance of stress [14].

Therefore, it is necessary to measure the state of stress in individuals. Furthermore, it is necessary to take steps to improve a bad stress state soon after it occurs [2]. Therefore, we must be able to assess a person's emotional state while considering the corresponding relations to biology, psychology, and actions for stressors of various kinds. The relations between the changing expression intensity and psychological stress with facial expressions can be verified from their psychological and behavioral characteristics. We can assess expressions of individual facial expressions using Facial Expression Spatial Charts (FESCs) [15], which were proposed as a new framework to describe facial expression spaces and patterns of expressive intensity constituting each facial expression. Facial expression spaces are spatial configurations of each facial expression that are used to analyze semantic and polar characteristics of various emotions portrayed by facial expressions [4]. They represent a correspondence relation between the physical parameters that present facial changes expressed by facial expressions and the psychological parameters that are recognized as emotions. Our experiment results suggest the influence of psychological stress on facial expressions. For this study, we create a model of stress elements for individuals of both genders using BNs. Then, we graphically analyze the interdependence between psychological stress and facial expressions. For this study, we designate a parameter of expression intensity that characterizes facial expressions quantitatively.

IV. DATASETS

For this study, we constructed a dataset to assess facial expression changes. We measured the psychological stress of a subject showing facial expression changes using SRS-18. Then we compared the results to the facial expressions.

A. Facial Expression Images

We set the time period during which we measured facial expressions to construct individual models of stress elements. We constructed an original and long-term dataset for the specific facial expressions of one subject. For the experiment, we created original facial expression datasets from 10 subjects, with each dataset including images with three facial expressions: happiness, anger, and sadness obtained at one-week intervals during 7-20 weeks. The subjects were five women (Subjects A, B, C, and D were 19; Subject E was 21) and five men (Subjects F and J were 19; Subjects G, H, and I were 22), all of whom were university students. The order of facial expressions in a single measurement is happiness→anger→sadness. When taking images of each facial expression, the same expression is repeated three times based on neutral facial expressions during the image-taking time of 20 s. We previously instructed subjects to express an emotion three times during the image-taking time. One set of data consisted of 200 frames with the sampling rate of 10 frames per second.

We set the region of interest (ROI) to 90 × 80 pixels, including the eyebrows, which all contribute to the impression of a whole face as facial feature components. We set the ROI of the upper part to 40 x 80 pixels including the eyebrows, which contribute to the impression of upper facial parts as facial feature components. We set the ROI of the lower part to 50 × 80 pixels including the mouth, which contributes to the impression of lower facial parts as facial feature components.

B. Target Facial Expressions

We set three facial expressions of object facial expressions because the facial expressions are acquired over a long term for our study. We strove to reduce the load on subjects. We selected happiness, anger, and sadness from six basic facial expressions described by Ekman [16]. He asserted that Japanese people show disgust by smiling to conceal their true emotions. Therefore, we consider that it is difficult for subjects participating in this study to express disgust. The emotion of fear is a rare feeling in daily life. In the opinion of subjects, it was often stated that they are not aware of how they appear when feeling fear. Therefore, we do not record fear among the facial expressions. Surprise can readily occur along with fear, happiness, solace, anger, and despair [16]. Therefore, we do not include surprise among the recorded facial expressions because it invariably translates into complex facial expressions.

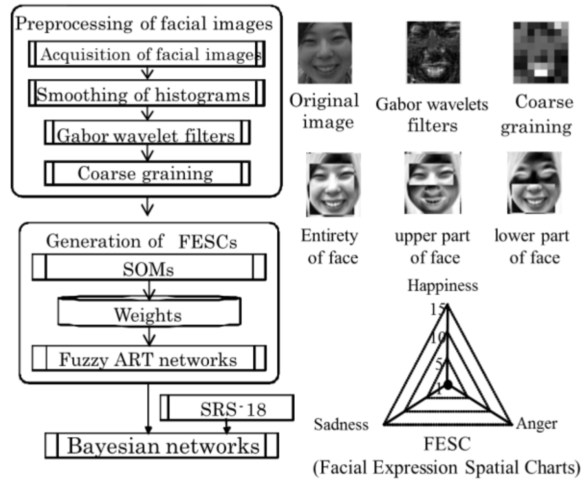


Figure 2. Procedure of the proposed method.

Our target facial expressions are therefore happiness, anger, and sadness, which include the geometry of each quadrant of Russell's circumplex model [17].

C. Stress Measurement

Psychological stress reactions are anxiety, anger, lassitude and difficulty in concentrating, which are encountered on a daily basis when one is affected by stressors. Our measurements are three robust stress factors: "depression and anxiety", "displeasure and anger", and "lassitude". Subjects respond on the check sheet using four responses for 18 items, answering with responses from "Completely different" to "It's correct". Each answer receives a score of 0–3. High point totals signify a higher degree of stress. Moreover, stress levels are represented by consultation value Level 1 (weak), Level 2 (normal), Level 3 (slightly high), and Level 4 (high). For this paper, we define the reported values as stress levels. For this experiment, we measured stress values using the SRS-18 and took facial expressions of 10 subjects. To avoid influencing the facial expressions, we did not report any score to subjects. Moreover, subjects recorded their responses in accordance to the SRS-18 scale before recording facial expressions.

D. Extraction of Facial Expressive Intensity

Figure 2 portrays a flow chart of our proposed method. Features are emphasized using Gabor wavelet filters in the preprocessing of input images. For extracting and normalizing the topological variation of facial expressions,

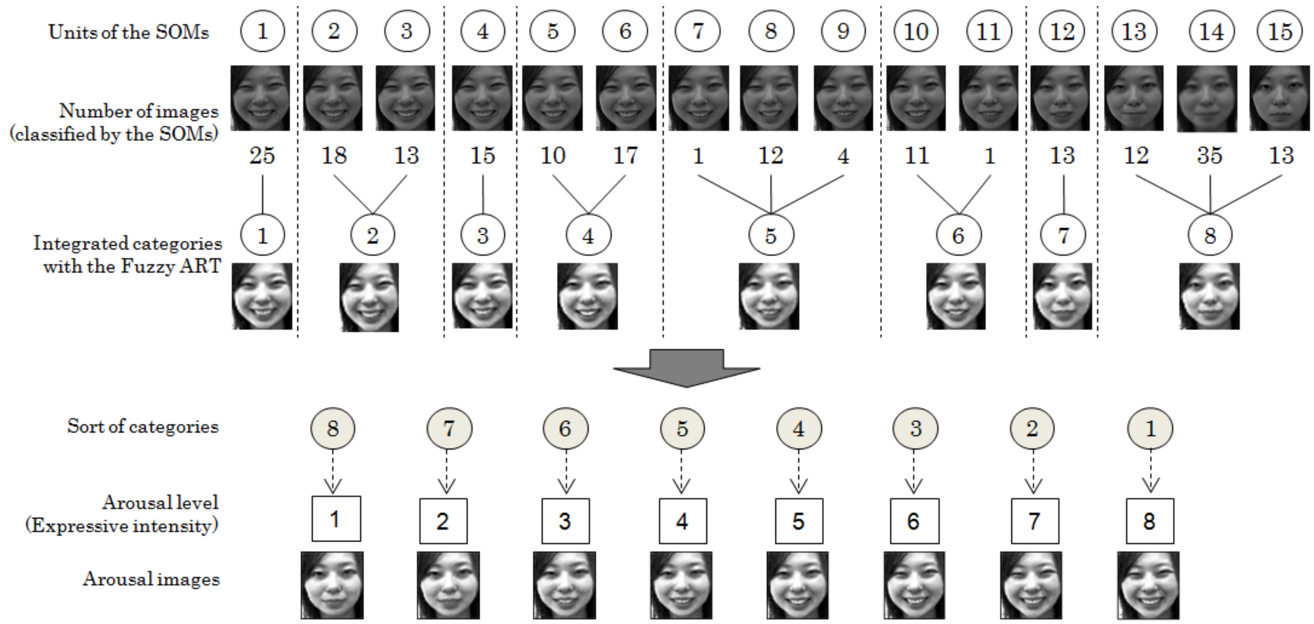


Figure 3. Expressive intensity.

we reduce noise and constrain the amount of information using time-series images decorated with Gabor wavelets and processing coarse graining. The period during which images were obtained was expanded from several weeks to several months. Although we took facial expression images in constant conditions, we were unable to constrain external factors completely, e.g., through lighting variations. Therefore, in the first step, brightness values are preprocessed with normalization of the histogram to the target images. In the next step, features are extracted using Gabor wavelet filtering. The information representation of Gabor wavelets that can emphasize an arbitrary feature with inner parameters shows the same characteristics of response selectivity in a receptive field. Gabor wavelets are functions that combine with a plane wave propagating to one direction and a Gaussian wave. We used the three directions of 0, 45, and 90 deg for selective responses for facial parts. At the final step, we applied down-sampling for noise reduction and data size compression. For this method, we set the initial template position to include facial parts for capturing facial images. We use template-matching methods to trace the region of interest of a face in real time. However, the trace results of the region of interest yield errors caused by body motion. These errors can be removed through down-sampling. The down-sampling window that we set is 10 × 10 pixels. The target images are compressed from 80 × 90 pixels to 8 × 9 pixels.

Facial expression processes differ among individuals. Therefore, Akamatsu described the adaptive learning mechanisms necessary for modification according to individual characteristic features of facial expressions. This study targets intentional facial expressions. We use SOMs [18] to extract topological changes of facial expressions and for normalization with compression in the direction of the temporal axis. After classification by SOMs, facial images are integrated using Fuzzy ART [19], which is an adaptive learning algorithm with stability and plasticity. In fact, SOMs perform unsupervised classification input data into a mapping space that is defined preliminarily. In contrast, Fuzzy ART performs unsupervised classification at a constant granularity that is controlled by the vigilance parameter. Therefore, using SOMs and Fuzzy ART together, time-series datasets showing changes over a long term are classified with a certain standard. Here, we used one-dimensional SOMs with 15 mapping units and set the vigilance parameter of Fuzzy ART to 0.90 through a preliminary experiment.

Figure 3 presents details of procedures for acquiring a time-series variation of expressive intensity. First, we use SOMs to learn the time-series images of facial expressions with down-sampling. The face images that show topological changes of facial expressions that are similar are classified into 15 mapping units of SOMs. Next, similar units among 15 mapping units of SOMs are integrated into the same category by Fuzzy ART. By sorting the facial expression categories integrated by Fuzzy ART from neutral facial expression to the maximum of facial expression, we obtain categories labeled as expressive intensities of facial expressions quantitatively.

V. MODEL OF STRESS ELEMENTS

A Bayesian network is a knowledge representation scheme dealing with probabilistic knowledge [20]. Its nodes and arcs mutually connect, forming a directed acyclic graph. Each node can be viewed as a domain variable that can take a set of discrete values or continuous value. An arc represents a probabilistic dependency between the parent node and the child node. We illustrate the graphical modeling approach using a real-world case study, such as modeling and inferring human psychological stress by integrating information from intentional facial expressions and four levels of expression intensity, three stress factors, and 18 stress attributions of SRS-18. A probabilistic psychological stress model based on the BNs is a suitable option to address the relation between facial expressions and human psychological stress.

Table I presents parameters representing the psychological state, which is the input of the stress element model, i.e., three stress factors and 18 stress elements. Table II shows the state values of three facial expressions (i.e., "happiness", "sadness", and "anger") and the stress response degree as stochastic variables for outputting occurrence probabilities. In the BNs, the state value of each variable, the conditional probability, and the graph structure are constructed by statistical learning using the previously obtained dataset. The constructed BNs determined the state value of the expression type as an output variable from the observed dataset. Then, it performed probabilistic inference for the parameters representing the psychological states of the input variables. Consequently, the probability distribution of each state value of the input variable is obtainable. The BNs might be assessed as stochastic parameters representing the mental state at that time from the occurrence probability distribution of intentional facial expressions. Therefore, using Bayesian networks, we were able to estimate the parameters determining the state of mind at that time and specific the type of facial expression easy to expose at a certain psychological state.

Using a program package of Bayesian Network Construction System (BayoNet) [21], the stress element models used for this study were constructed with the structure learning employing K2 algorithm [22] as a model search algorithm, Akaike Information Criterion (AIC) [23] as the model evaluation criterion. The details are explained below.

A. Definition of Variable Nodes

The stress model used in our experiment comprises 25 nodes. A stress element model was constructed from 18 stress elements, three stress factors, three facial expression intensities, and one stress response degree. The "depression and anxiety", "displeasure and anger", and "lassitude" of stress factors accommodate parent nodes for the six stress elements. The stress factors are parent nodes to the stress response degree node, which are the child nodes. We set relations manually between the parent and child of facial expression intensities and stress factors. We set stress factors and expression intensity respectively as parent and child nodes based on preconditions for psychological stress influence to facial expressions. Furthermore, we set directed links for stress elements as six nodes to each stress factor based on a precondition that stress elements trigger stress factors.

The 18 nodes of stress elements are assigned 0–3 points with four items: "Strongly no", "Yes a little", "Yes", and "Definitely yes". All stress factor nodes have four grades: Level 1 (weak), Level 2 (normal), Level 3 (slightly high), and Level 4 (high).

TABLE I. INPUT VARIABLES OF BNS

Stress factor	State	Stress element	State
Depression and anxiety	Level 1 (i.e., weak) Level 2 (i.e., normal) Level 3 (i.e., slightly high) Level 4 (i.e., high)	sad mood	3:Definitely yes 2:Yes 1:Yes a little 0:Strongly no
		worry	
		feel like crying	
		disheartened	
		being bored	
		want to comfort	
Displeasure and anger	Level 1 (i.e., weak) Level 2 (i.e., normal) Level 3 (i.e., slightly high) Level 4 (i.e., high)	quick-tempered	3:Definitely yes 2:Yes 1:Yes a little 0:Strongly no
		feel anger	
		lose one's cool	
		feel unhappy	
		be annoyed	
		feel frustrated	
Lassitude	Level 1 (i.e., weak) Level 2 (i.e., normal) Level 3 (i.e., slightly high) Level 4 (i.e., high)	have no confidence	3:Definitely yes 2:Yes 1:Yes a little 0:Strongly no
		consider bad things	
		not completed	
		want to be alone	
		can't focus	
		not patience	

TABLE II. OUTPUT VARIABLES OF BNS

Facial expression	Region of interest	State
Happiness	Entirety of face	1 – 15 (Expressive intensity)
	Upper part of face	
	Lower part of face	
Anger	Entirety of face	1 – 15 (Expressive intensity)
	Upper part of face	
	Lower part of face	
Sadness	Entirety of face	1- 15 (Expressive intensity)
	Upper part of face	
	Lower part of face	
Stress response degree		Level 1 (i.e., weak) Level 2 (i.e., normal) Level 3 (i.e., slightly high) Level 4 (i.e., high).

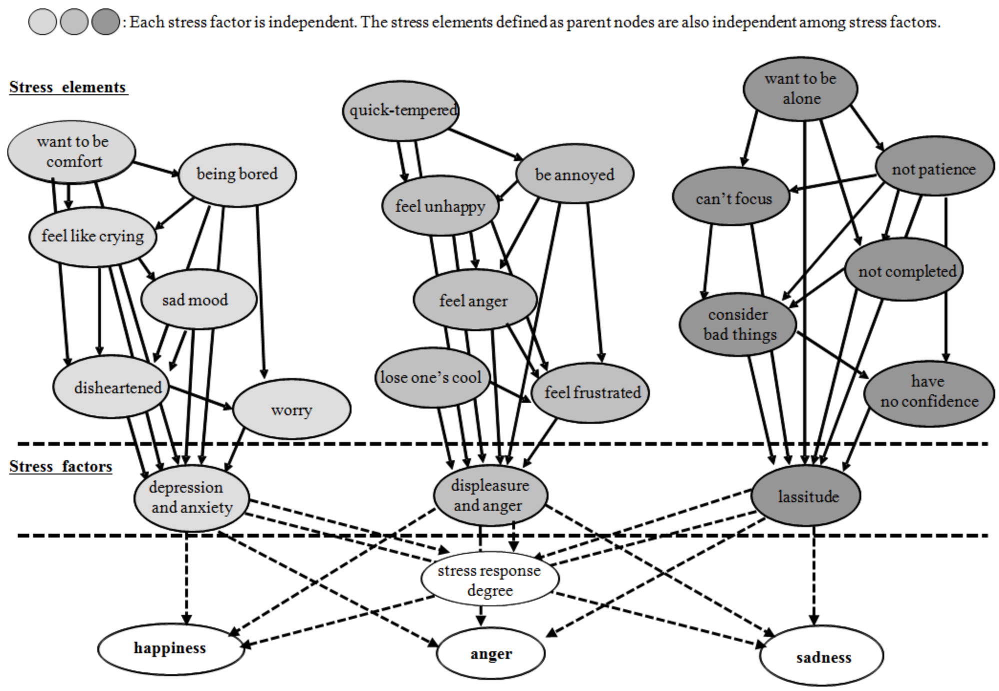


Figure 4. Stress elements model without constraints.

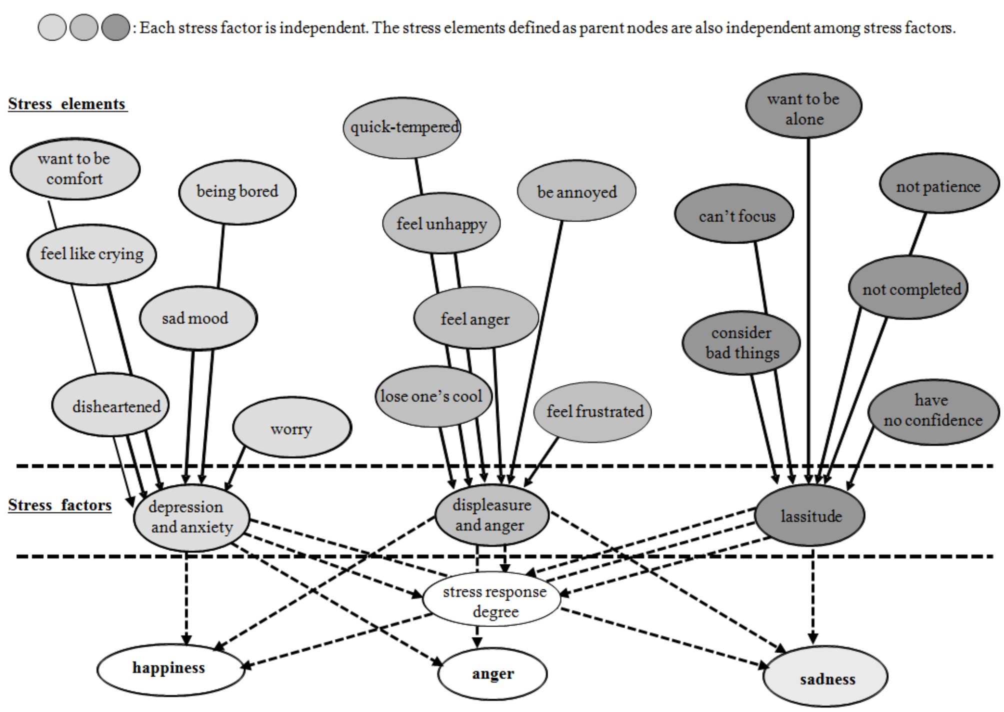


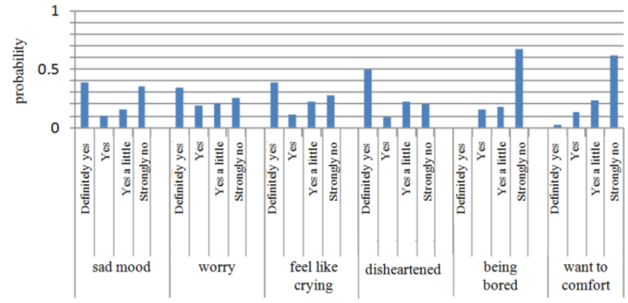
Figure 5. Stress elements model with constraints.

VI. Analysis of Stress Element Model

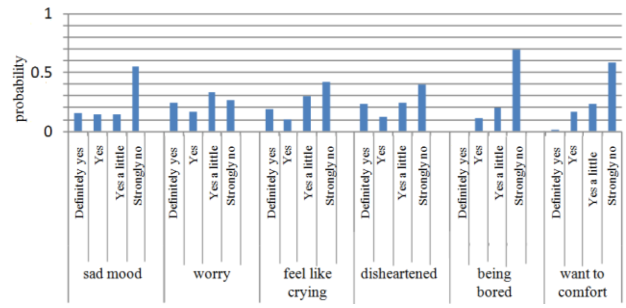
For the analysis described in this section, we constructed two models given certain constraints on different nodes of stress factors to obtain a single simple model for which effects of stress are noticeable. Using the selected model, we attempt to analyze the types of facial expressions that are readily influenced by stress. Subsequently, we shall analyze the stress factors and stress elements supporting them. To ascertain differences in stress susceptibility by gender, we compare the results of analysis of stress elements obtained for men with those obtained for women.

A. Model Construction

As a preliminary experiment, we constructed models of stress elements under two constraints: "with constraints, where each factor is independent"; and "without constraints, allowing relations among elements". The model of "without constraints" in Figure 4 is constructed with an optimal graph structure representing the correlation of multiple stages by connecting each stress element with an arc. However, the model of "with constraints" in Figure 5 shows a simple graph structure by which stress elements form directed links correspond to stress factors. As might be inferred from the contents of the preliminary experiment, no need exists to define an exact correlation between stress elements because a tendency is apparent by which the probability distributions of the stress levels and expressive intensities are similar, irrespective of the presence or absence of constraints. However, the probability distribution of the model without constraints is an important characteristic for the analysis of stress elements. In other words, although the stress elements that characterize the factor of "depression and anxiety" are not found in Figure 6(b), dominant changes appear in the probability distribution of "sad mood", "feel like crying" and "disheartened" as stress elements characterizing the factor of "depression and anxiety" in Figure 6(a). It is regarded as more effective when performing probabilistic inference.



(a) Without constraints



(b) With constraints

Figure 6. Probability distribution of stress elements.

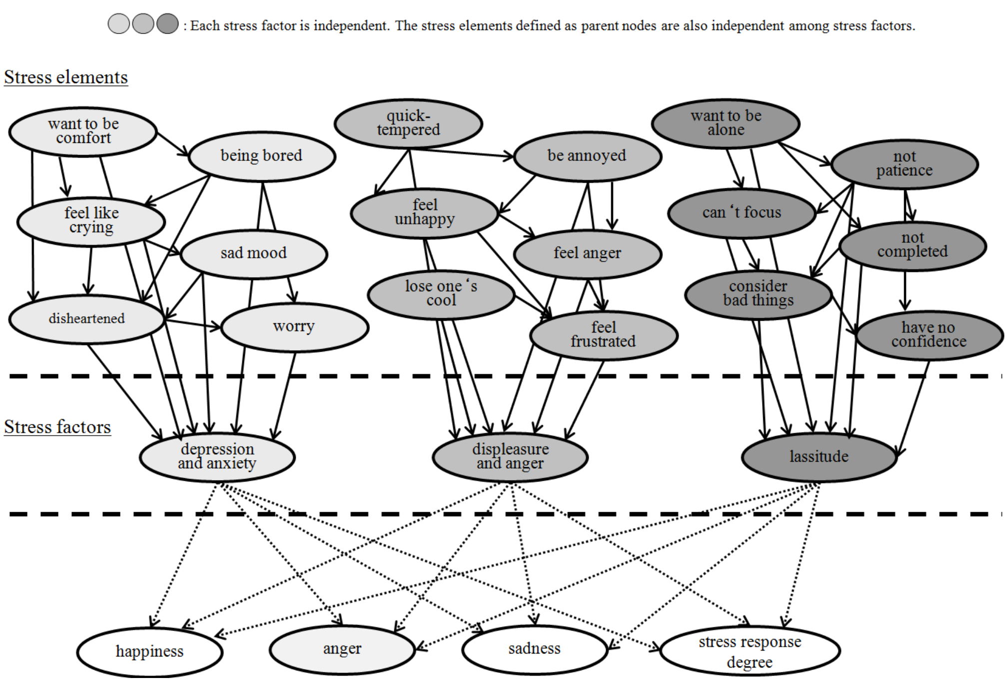


Figure 7. Stress elements model of male.

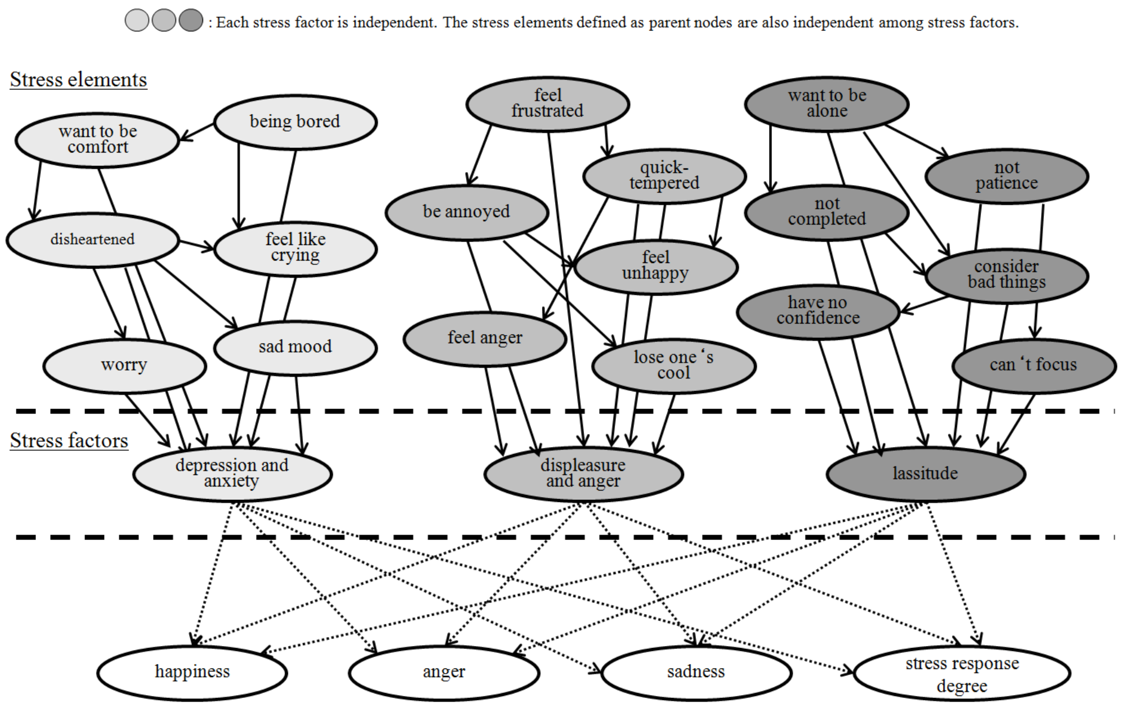


Figure 8. Stress elements model of female.

Therefore, for the following experiments, we use the stress element model without constraints, allowing their relation among elements.

B. Analytical Procedures

The stress element models of men and women are presented in Figure 7 and Figure 8. Each node in stress factors, such as "depression and anxiety", "displeasure and anger", and "lassitude", has directed links to six items of stress elements as parent nodes. The directed links signify that nodes connected with them are optimized for better inferential accuracy. Therefore, a strong mutual relation exists between nodes that are connected by directed links.

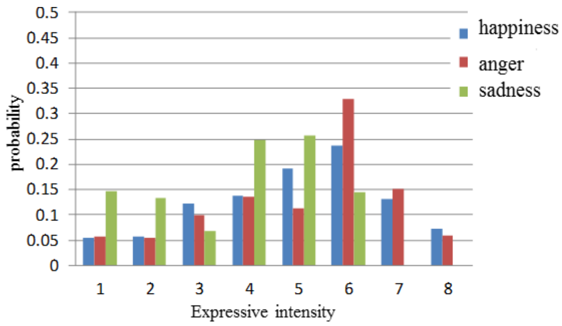
The analytical procedures of stress factors that affect the facial expressions are described in the following based on the stress element models. As the flow of the entire analysis, by giving evidence to the degree of stress response, and by comparing the probability distribution of the expressive intensities "happiness", "anger" and "sadness", we strive to identify facial expressions that are sensitive to stress. Additionally, we verify all stress elements in terms of whether they affect the expressive intensities of the facial expression.

First, we calculate the expressive intensities of three facial expressions by giving evidence to the stress response degree as "weak". Conducting stochastic reasoning based on the probability distribution related to each expressive intensity, the most sensitive stress that is likely to appear in any facial expression was determined. Next, we assessed the degree of stress responses of "slightly high" and "normal" using the same procedures. We specifically examine the probability distribution of expressive intensities in three facial expressions. Furthermore, even for stress elements corresponding to each stress factor, using the same procedures as stochastic reasoning, we identify the stress factors and elements that affect expressive intensities in facial expressions.

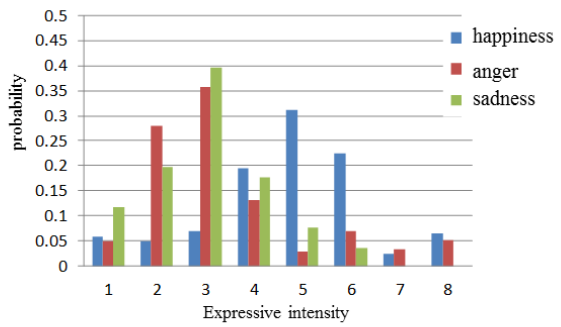
C. Male Model

Figure 9 presents a probability distribution of the expressive intensities corresponding to each stress response degree in the male model. Giving evidence to the stress response degree as "weak", the probability value was also larger as the expressive intensity increases. For the case in which the stress response degree is "normal" compared to "weak", the probability value is greater when the expressive intensity is small. Moreover, in the case of "slightly high", probability values tend to be large in all three facial expressions, when expressive intensities are small. Estimating the expressive intensity corresponding to stress response degree is particularly difficult for facial expressions of "anger" and "sadness" because it is small in the state of "normal" stress.

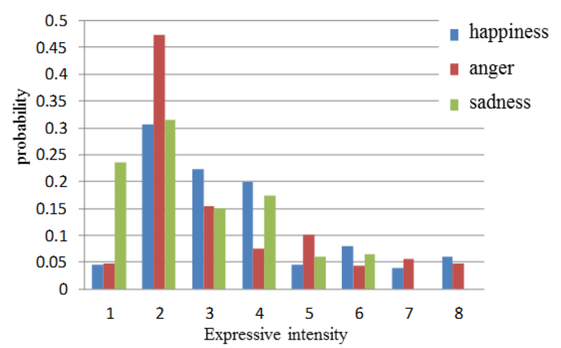
Therefore, we will strive to conduct analyses particularly addressing the "happiness" facial expression, for which the probability value of expressive intensity is changing related to the stress response degree. To analyze the stress factors and stress elements that support them on the "happiness" facial expression, by giving evidence sequentially from "Level 1" to "Level 4" for total stress levels, we identify the



(a) Stress response degree: weak



(b) Stress response degree: normal

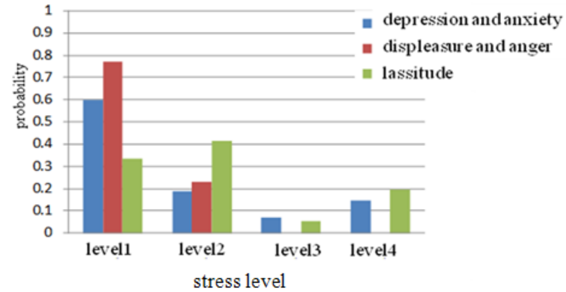


(c) Stress response degree: slightly high

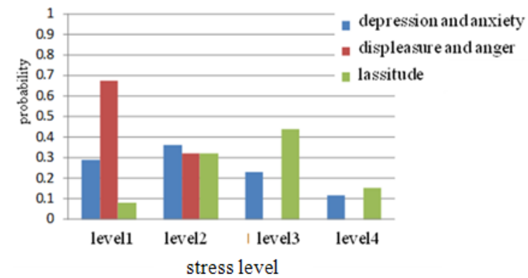
Figure 9. Probability distribution of the expressive intensities corresponding to each stress response degree in male model.

stress factors affecting facial expressions from their probability distributions of expressive intensities. Furthermore, we examine the link structure of the stress elements related to the stress factor that has been identified.

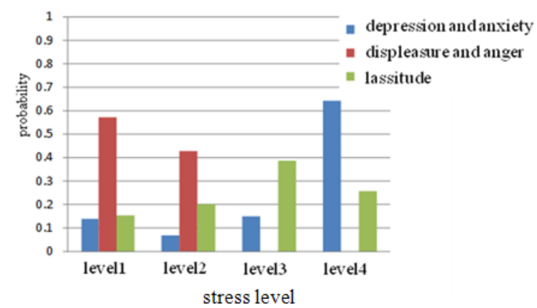
Figure 10 depicts a probability distribution of the stress level in each stress factor. As the stress response degree becomes higher, the level of the stress factor "depression and anxiety" indicates a higher value. This phenomenon appears most significantly on the estimated value of the probability distribution. However, for the stress factors of "lassitude" and "displeasure and anger", characteristic changes are not observed in the estimated value of the probability distributions at respective stress levels.



(a) Stress response degree: weak



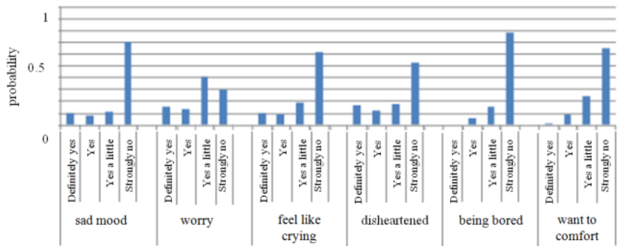
(b) Stress response degree: normal



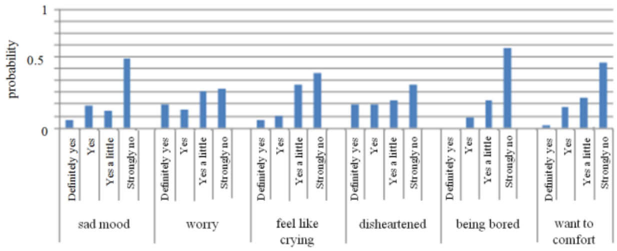
(c) Stress response degree: slightly high

Figure 10. Probability distribution of the stress levels with emphasis on the facial expression of "happiness".

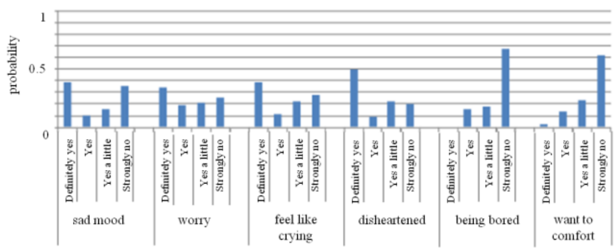
Therefore, in the male model, we consider that the factor of "depression and anxiety" affects the facial expressions of "happiness" as a stress factor. Then, targeting the six stress elements characterizing the factors of "depression and anxiety", we attempt to identify the stress factors for supporting the relevant factor. We set evidence for the stress response degrees of "normal", "weak", and "slightly high". Figure 11 portrays the probability distribution of stress elements for each stress level. For "normal" and "weak" as stress response degrees, the highest probability value of "Strongly no" denying the stress elements was identified. Other probability values were less than 0.3. For "slightly high" as a stress response degree, stress elements of "sad mood", "feel like crying", and "disheartened" showed high probability values together for support of the cause of "Definitely yes".



(a) Stress response degree]e : weak

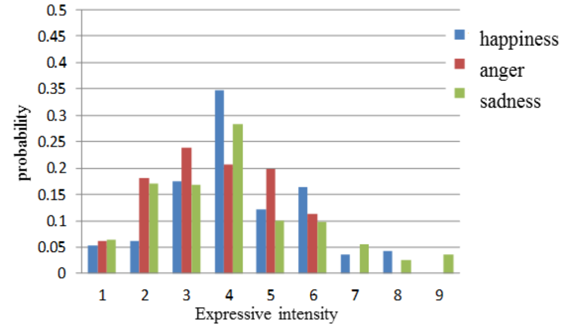


(b) Stress response degree : normal

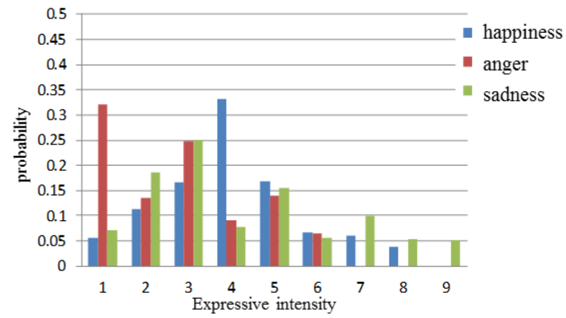


(c) Stress response degree : slightly high

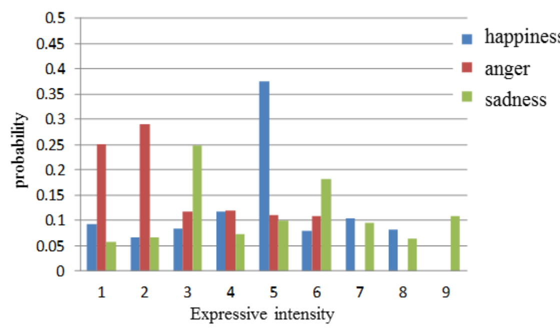
Figure 11. Probability distributions of the stress elements characterizing the factors of "depression and anxiety" in male model.



(a) Stress response degree : weak



(b) Stress response degree : normal



(c) Stress response degree : slightly high

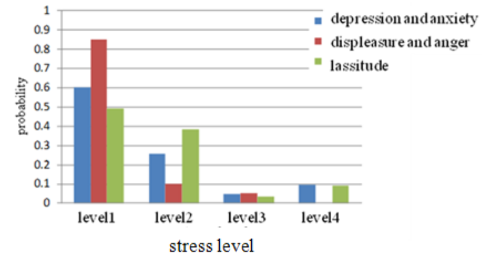
Figure 12. Probability distribution of expressive intensities corresponding to each stress response degree in female model.

D. Female Model

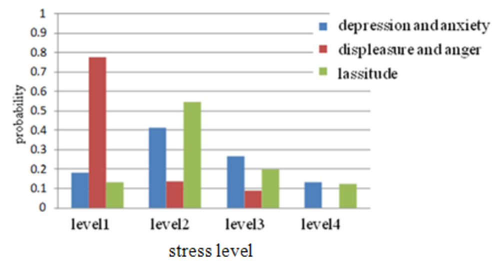
Using the same procedure as that used for men, we analyzed the relations among stress factors, stress elements, and expressive intensities of three facial expressions for women. In the female model, a marked change was recognized in the probability distribution of expressive intensity of "sadness", as presented in Figure 12.

Figure 13 presents a probability distribution of stress levels with emphasis on the facial expression of "sadness" in each stress factor. As might be understood from the contents of Figure 13, particularly addressing the probability distribution of the factors of "lassitude" and "depression and anxiety", a considerable change is apparent with the difference of stress response degree, i.e., "weak", "normal", and "slightly high". The probability distributions of the stress elements characterizing the factor of "depression and anxiety" are presented in Figure 14. As a stress element giving influence to the factor of "depression and anxiety", the influence of "disheartened" exists to a slight degree. However, no distinctive element to support the factor of "lassitude" is found anywhere because the probability distribution shows a similar tendency to that of the change of the degree of stress response.

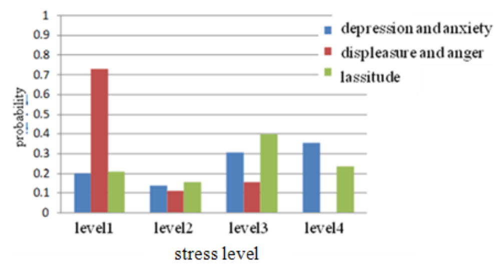
Based on the experimentally obtained results presented above, we conduct an examination from the perspective of stress factors and stress elements giving influence to them, specifically examining the relation between psychological stress and three facial expressions for men and women. Some facial expressions appear easily, but others are difficult to assess in terms of psychological stress. In the male model, facial expressions of "happiness" show marked changes attributable to differences in the stress response degree, but characteristics of the probability distribution of expressive intensities are similar in facial expressions of "sadness" and "anger". Expressive intensities become slight with the increase of stress response degree as an overall trend. In the female model, a change was observed in the



(a) Stress response degree : weak

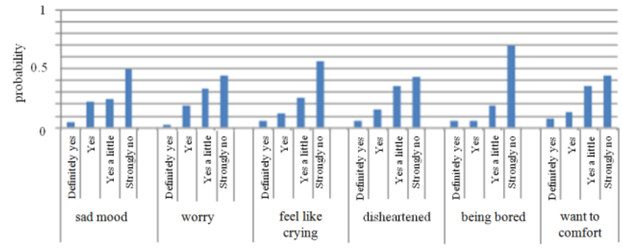


(b) Stress response degree : normal

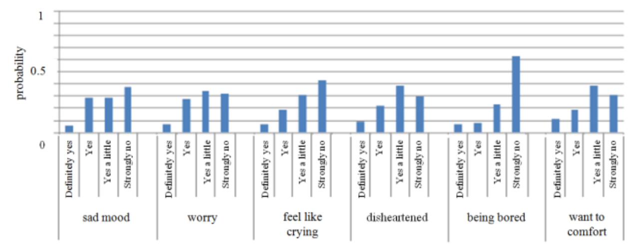


(c) Stress response degree : slightly high

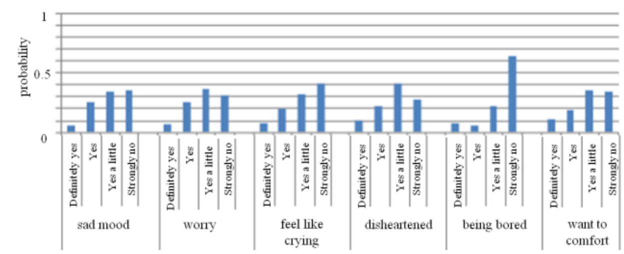
Figure 13. Probability distribution of stress levels with emphasis on the facial expression of "sadness".



(a) Stress response degree : weak



(b) Stress response degree : normal



(c) Stress response degree : slightly high

Figure 14. Probability distributions of the stress elements characterizing the factors of "depression and anxiety" in female model.

characteristics of the probability distribution of expressive intensities, only the facial expression of "sadness", by setting evidence to the stress response degree as "slightly high".

Therefore, we inferred that the influence of psychological stress appears easily in the expression of "happiness" for men and in the expression of "sadness" for women. In addition, the factor of "depression and anxiety" as a stress factor influences the facial expressions of "happiness". Then, as stress elements which support it, three items exist for men: "sad mood", "feel like crying", and "disheartened". Although significant changes were observed in the probability distribution of the factors of "lassitude" and "depression and anxiety", the female model did not engender specific stress elements to support those stress factors.

VII. Stress Factors and Effect of Face Region

In this section, we first classify the state (slightly high) of levels 3–4 and the state (weak) of level 1 for which each stress factor holds the state of "Level 4" from "Level 1". Next, by probabilistic reasoning of giving evidence in these two states, we strive to identify the face region (upper part of the face, lower part of the face) in which psychological stress effects readily appear. For this experiment, we use the model of stress elements for all subjects: the 10 university student subjects comprise 5 men and 5 women. The stress element model of all subjects is presented in Figure 15. This model is a graph structure consisting of 28 nodes: 18 nodes corresponding to the stress elements, 3 nodes corresponding to the stress factors, one node corresponding to the stress response degree, and 6 nodes corresponding to the expressive intensities appearing in the lower part and upper part of the face on three facial expressions.

A. *Factor of 'Depression and Anxiety'*

Figure 16 presents expressive intensities of parts of the face, where we set evidence to the state (slightly high) of levels 3–4 and the state (weak) of level 1, assessing the factor of "depression and anxiety". Estimation results of stress levels and expressive intensities are presented in Figure 16(a). The relation between types of facial expression and the differences of expressive intensity are presented in Figure 16(b). These two figures are summaries

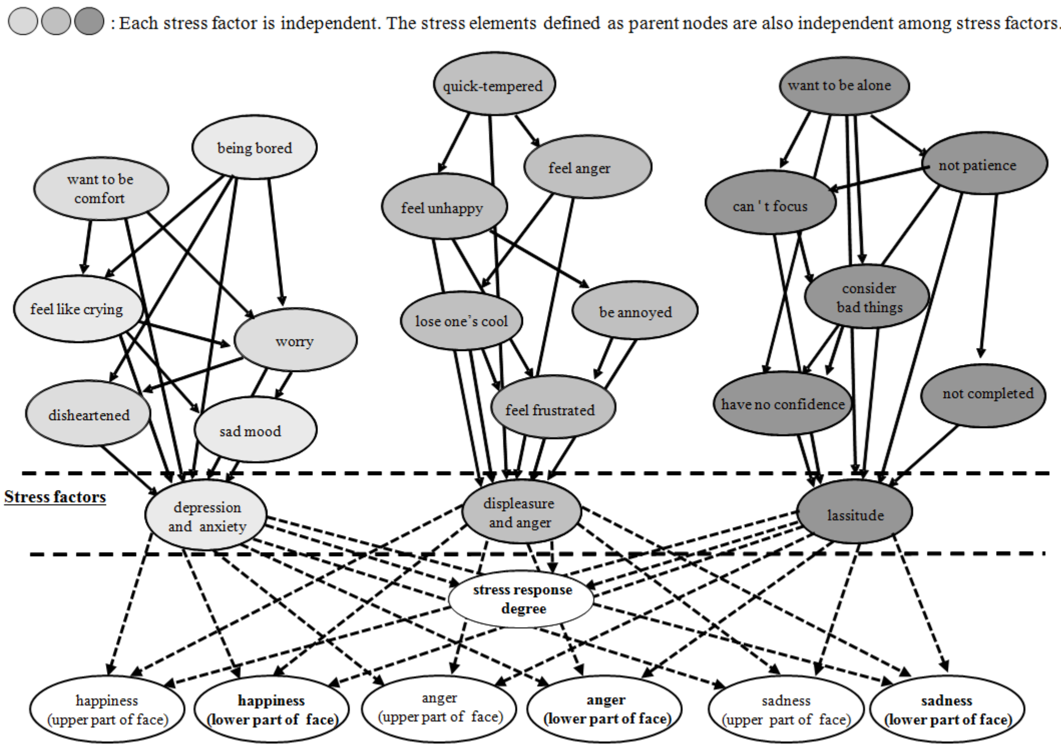


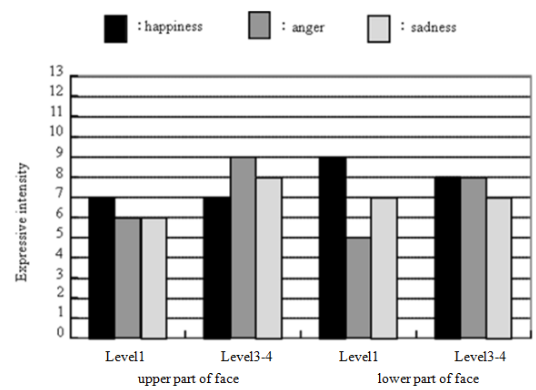
Figure 15. Stress elements model of all subjects.

for the respective face regions. The vertical axes in the figures respectively show the expressive intensity and the difference of expressive intensity. It is noteworthy that the difference of expressive intensity shows the absolute value of the difference between the expressive intensity levels "3-4" and "level 1" in respective states.

Specifically, assessing the differences of expressive intensity in "happiness", "anger", and "sadness", the respective values of the upper part of the face are 0, 3, and 2. In contrast, the respective values of the lower part of the face are 1, 3, and 0. In the case in which the stress factor of "depression and anxiety" acts significantly, large values have been identified for the upper and the lower part of the face at expressing "anger". Therefore, the influence of "depression and anxiety" readily appears at expressing 'anger'. Additionally, we infer that the influence affects the entire face region. These analytical results are consistent with the contents of the previous study described in [18], i.e., the emotional states of mind stimulating "anger" and "depression" are similar.

B. *Factor of 'Displeasure and Anger'*

Figure 17 presents the expressive intensities of each face region based on the stress factors of "displeasure and anger", as presented in the preceding section. Specifically examining the differences in expressive intensity in "happiness", "anger", and "sadness", the respective values of the upper part of the face were 2, 1, and 4. In contrast, the



(a) Estimation of stress levels

(b) Difference of expressive intensity each face part

Figure 16. Expressive intensities of parts of the face.

respective values of the lower part of the face are 0, 1, and 2. Accordingly, we consider that the following analysis is reasonable, i.e., the influence of "displeasure and anger" readily appears at the upper part of the face of expressing "anger".

In addition, because we are conducting the operation of "glaring" to intimidate an opponent expressing "anger", this analytical result matches a case study showing changes that occur during that process, such as "eyebrows down" or "upper eyelid is raised". Consequently, the effect readily appears strongly in the upper part of the face of the facial expression "anger", for the case in which the stress factor of "displeasure and anger" is readily apparent.

C. *Factor of 'Lassitude'*

Figure 18 exhibits expressive intensities of respective face regions based on the stress factor of "lassitude". Specifically examining the differences of expressive intensity in "happiness", "anger", and "sadness", the respective values of the upper part of the face are 5, 3, and 2. In contrast, the respective values of the lower part of the face are 1, 3, and 4. Accordingly, we regard the following analysis as reasonable: the influence of "lassitude" readily appears at the upper part of the face expressing "happiness", and the lower part of the face expressing "sadness".

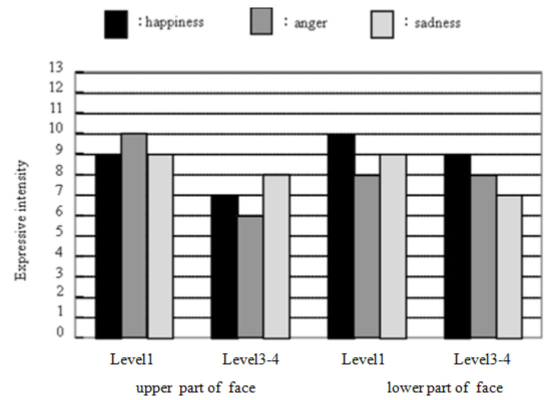
In general, changes in facial expressions are poor during the state of "lassitude". Therefore, a tendency to fall "expressionless" appears to be confirmed. However, results indicate that the influence appears strongly in the lower part of the face for the expression of "sadness" and in the upper part of the face of the expression 'happiness' in this analysis. For this study, we adopted an experimental protocol in which all subjects intentionally express three facial expressions of "happiness", "sadness", and "anger", even when they are in the state of "lassitude". Therefore, these reasoning results reflect the characteristics of datasets based on the experimental protocol. Although the factor of "lassitude" contributes significantly, results show that the effect readily appears in the lower part of the face of the expression "sadness" and in the upper part of the face of the expression "happiness" for intentional facial expressions.

VIII. CONCLUSION AND FUTURE WORK

This study, which analyzed stress factors and elements of psychological stress on intentional facial expressions using BNs, was conducted to identify the types of facial expressions and facial parts which readily manifest the influence of psychological stress. Our evaluation experiment used stochastic reasoning to build stress element models for male and female subjects, providing evidence for the stress response degree and stress factors.

Results revealed the following points.

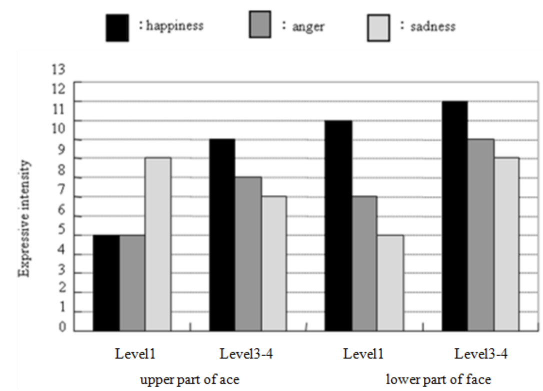
1) When building a stress element model, a model of "without constraints" that allows relations among stress elements is more beneficial for analysis of probabilistic reasoning results than a model of "with constraints".



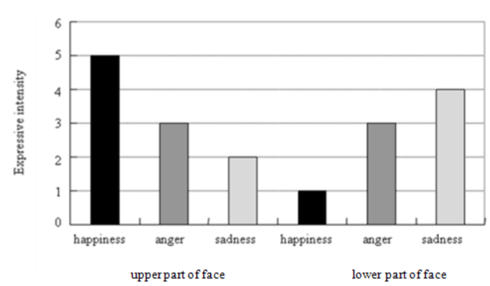
(a) Estimation of stress levels

(b) Difference of expressive intensity each face part

Figure 17. Expressive intensities of each face region based on the stress factors of "displeasure and anger".



(a) Estimation of stress levels



(b) Difference of expressive intensity each face part

Figure 18. Expressive intensities of respective face regions based on the stress factor of "lassitude".

2) The "depression and anxiety" factor affects the facial expression of "happiness" in men.
3) The "depression and anxiety" and "lassitude" factors affect the facial expressions of "sadness" in women.
4) The influences of "depression and anxiety" readily appear when expressing "anger". They affect the entire face.
5) The influence of "displeasure and anger" readily appears in the upper part of the face when expressing "anger".
6) The influence of "lassitude" strongly appears in the lower part of the face of the expression of "sadness", in the upper part of the face of the expression of "happiness".

We will estimate models of stress elements by increasing the number of subjects and the image capture duration. Furthermore, we will improve our method to address both intentional facial expressions and natural expressions that are exposed unconsciously.

Acknowledgment

This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 25330325.

References

[1] K. Sato, H. Otsu, H. Madokoro, and S. Kadowaki, “Analysis of psychological stress factors and facial parts effect on intentional facial expressions,” Proceedings of The Third International Conference on Ambient Computing, Applications, Services and Technologies (AMBIENT2013), pp. 7-16, Oct. 2013.

[2] T. Yata, H. Sannohe, M. Nakasako, and M. Tao, “How to cope with stress,” Yuhikaku, 1993.

[3] S. Akamatsu, “Recognition of facial expressions by human and computer [I]: facial expressions in communications and their automatic analysis by computer,” The Journal of the Institute of Electronics, Information, and Communication Engineers, vol. 85, no. 9, pp. 680-685, Sep. 2002.

[4] H. Shimomura, K. Kanamori, J. Nishimaki, and K. Shiba, “Usefulness of salivary amylase and cortisol measurement as stress markers at educational sites,” ISSN, vol. 33, no. 3, pp. 247-254, 2010.

[5] N. Takai, M. Yamaguchi, T. Aragaki, K. Eto, K. Uchihashi, and Y. Nishikawa, “Effect of psychological stress on the salivary cortisol and amylase levels in healthy young adults,” Arch Oral Biol., vol. 49, no. 12, pp. 963-968, 2004.

[6] Medicore Co. Ltd., Body Checker (Cardio Monitor), http://www.medi-core.com/ [retrieved: June, 2014]

[7] M. Malik, “Heart rate variability: standards of measurement, physiological interpretation, and clinical use,” European Heart Journal, vol. 17, pp. 354-381, 1996.

[8] Brain Function Research Center, Alphatec-IV, http : //www.alphacom.co.jp/ [retrieved: June, 2014]

[9] R. S. Lewis, N. Y. Weekes, and T. H. Wang, “The effect of a naturalistic stressor on frontal EEG asymmetry,” Biological Psychology, vol. 78, pp. 239-247, 2007.

[10] I. Akirav and M. Maroun, “The role of the medial prefrontal cortex–amygdala circuit in stress effects on the extinction of fear,” Neural Plast. 2007; 2007: 30873. Published online 2007 Jan. doi: 10.1155/2007/30837.

[11] D.M. McNair, J.W.P. Heuchert, and E. Shillony, Research with the Profile of Mood States (POMS), Toronto, Canada: Multi-Health Systems, 1964-2001.

[12] Comprehensive Support Project, ISTRESS Scale Guidebook, Jitsumu Kyouiku, 2004.

[13] S. Suzuki, “Stress Response Scale-18,” Kokoronet, Jul. 2007.

[14] Mechanical Social Systems Foundation, “Surveillance Study to Security Precaution of Stress Instrumentation Technology on Possible Application,” 2004.

[15] H. Madokoro, K. Sato, and S. Kadowaki “Facial expression spatial charts for representing time-series changes of facial expressions,” Japan Society for Fuzzy Theory, vol. 23, no. 2, pp. 157-169, 2011.

[16] P. Ekman, “Emotions Revealed: Understanding Faces and Feelings,” Kawade Shobo Shinsha, 2004.

[17] J.A. Russell and M. Bullock, “Multidimensional scaling of emotional facial expressions: similarity from preschoolers to adults,” Journal of Personality and Social Psychology, vol. 48, pp. 1290-1298, 1985.

[18] T. Kohonen, Self-organizing maps, Springer Series in Information Sciences, 1995.

[19] G.A. Carpenter, S. Grossberg, and D.B. Rosen, “Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system,” Neural Networks, vol. 4, pp. 759-771, 1991.

[20] Y. Motomura, “Probabilistic reasoning algorithms and their experiments in bayesian network,” IEICE, pp. 157-162, 2004.

[21] Y. Motomura, “Bayesian network software bayonet,” Journal of the Society of Instrument and Control Engineers, vol. 42, no. 8, pp. 693-694, 2003.

[22] G. Cooper and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data,” Machine Learning, no. 9, pp. 309-347, 1992.

[23] H. Akaike, “Information theory and an extension of the maximum likelihood principle,” Proceedings of the Second International Symposium on Information Theory, pp. 267-281, 1973.

REEF: Resolving Length Bias in Frequent Sequence Mining Using Sampling

Ariella Richardson
Industrial Engineering
Jerusalem College of Technology
Jerusalem, Israel
Email: richards@jct.ac.il

Gal A. Kaminka and Sarit Kraus
Computer Science
Bar Ilan University
Ramat Gan, Israel
Email: galk,sarit@cs.biu.ac.il

***Abstract*—Classic support based approaches efficiently address frequent sequence mining. However, support based mining has been shown to suffer from a bias towards short sequences. In this paper, we propose a method to resolve this bias when mining the most frequent sequences. In order to resolve the length bias we define *norm-frequency*, based on the statistical z-score of support, and use it to replace support based frequency. Our approach mines the subsequences that are frequent relative to other subsequences of the same length. Unfortunately, naive use of *norm-frequency* hinders mining scalability. Using *norm-frequency* breaks the anti-monotonic property of support, an important part in being able to prune large sets of candidate sequences. We describe a bound that enables pruning to provide scalability. Calculation of the bound uses a preprocessing stage on a sample of the dataset. Sampling the data creates a distortion in the samples measures. We present a method to correct this distortion. We conducted experiments on 4 data sets, including synthetic data, textual data, remote control zapping data and computer user input data. Experimental results establish that we manage to overcome the short sequence bias successfully, and to illustrate the production of meaningful sequences with our mining algorithm.**

***Index Terms*—Frequent Sequence Mining; Data Mining; Z-score; Sampling; Multivariate Sequences**

I. Introduction

In a previous study [1] we discussed resolving the length bias in frequent sequence mining. The frequent sequence mining problem was first introduced by Agrawal and Srikant [2] and by Mannila et al. [3]. There are many possible applications for frequent sequential patterns, such as DNA sequence mining [4], text mining [5], anomaly detection [6], [7], classification [8] and Web mining [9].

Frequent sequential pattern generation is traditionally based on selecting those patterns that appear in a large enough fraction of input-sequences from the database. This measure is known as *support*. In support based mining a threshold termed *minsup* is set. All sequences with a *support* higher than *minsup* are considered frequent.

Support based mining is known to suffer from a bias towards short patterns [10]: short patterns are inherently more frequent than long patterns. This bias creates a problem, since short patterns are not necessarily the most interesting patterns. Often, short patterns are simply random occurrences of frequent items. The common solution of lowering the *minsup* results in obtaining longer patterns, but generates a large number of useless short sequences as well [11]. Using confidence measures lowers the number of output sequences but still results in short sequences.

Thus, removing the short sequence bias is a key issue in finding meaningful patterns. One possible way to find valuable patterns is to add weights to important items in the data. Yun [12] provides an algorithm for frequent sequence mining using weights. The drawback of this technique is that for many data sets there is no knowledge of what weights to apply. Seno and Karypis [13] propose eliminating the length bias by extracting all patterns with a support that decreases as a function of the pattern length. This solution is based on the assumption that a short pattern must have a very high support to be interesting, and a long pattern may be interesting even with a lower support. Although this is a fair assumption in many scenarios, it is challenging to find a measure that can be used for frequent pattern mining without making an assumption on the relationship between frequency and length. Searching for closed or maximal patterns [14]–[16] is another way to approach this bias. However, mining closed or maximal patterns may not be the best approach to solve the short sequence bias. Using closed and maximal sequences ignores shorter partial sequences that may be of interest. Other approaches include comparing the frequency of a sequence to its subsequences [17], and testing for self sufficient sequences [18]. We propose an algorithm that mines sequences of all lengths without a bias towards long or short sequences. Horman and Kaminka [10] proposed using a normalized support measure for solving the bias. However, their solution is not scalable. Furthermore, they cannot handle subsequences that are not continuous or have multiple attributes. We allow holes in the sequence, for example: if the original sequence is ABCD, Horman and Kaminka can find the subsequences AB, ABC, ABCD, BC etc, but cannot mine ACD or ABD, whereas our proposed method can.

In [1], we presented an algorithm for **RE**solving l**E**ngth bias in **F**requent sequence mining (REEF), this algorithm is expanded in the current paper. REEF is an algorithm for mining frequent sequences that normalizes the support of each candidate sequence with a length adjusted z-score. The use of the z-score in REEF eliminates statistical biases towards finding shorter patterns, and contributes to finding meaningful patterns as we will illustrate. However, it challenges the scalability of the approach: z-score normalization lacks the anti-monotonic property used in support based measures, and thus supposedly forces explicit enumeration of every sequence in the database. This renders useless any support based pruning of candidate sequences, the basis for scalable sequence mining algorithms, such as SPADE [19].

In order to provide a means for pruning candidate sequences, we introduce a bound on the z-score of future sequence expansions. The z-score bound enables pruning in the mining process to provide scalability while ensuring closure. Details on how the bound is calculated will be described later in the paper. We use this bound with an enhanced SPADE-like algorithm to efficiently search for sequences with high z-score values, without enumerating all sequences. A previous preliminary study [20] indicates that this bound assists the speedup substantially, we add more proof of this in the current note. We use three text corpora, input from TV remote control usage and computer user input to demonstrate how REEF overcomes the bias towards short sequences. We also show that the percentage of real words among the sequences mined by REEF is higher than those mined with SPADE.

This paper enhances the previous work presented in [1] in several ways. First, we present the method used for the sampling of the data that we use for calculating the bound. We also present an extensive evaluation of the parameter setting for this method using the various data sets. Second, we report the runtime results for use of the bound and discuss them. Finally, we present the results of our experimental evaluation on two extra data sets, TV remote control usage (Zapping) and a synthetic data set, that were not reported in [1].

The structure of the paper is as follows: Section II describes the related work. Section III provides background and notation and introduces Norm-Frequent Sequence Mining Problem (with Sampling). In Section IV, the algorithm used for the Norm-Frequent Sequence Mining is described in detail. Experimental evaluation is provided in Section V, and finally Section VI concludes our paper.

II. Related Work

The topic of frequent sequence mining is highly researched. This essential data mining task has broad applications in many domains and is used for a variety of applications such as agent modeling. In multi-agent settings there are many usages to modeling agents. A group of coordinating agents must have a clear model of each agent in the group and agents working in adversarial environments must be able to model their opponent. In cases where there is no prior behavior library (as is often the case) it is necessary to use the observed behavior in order to deduce the model. This task can be performed using the multivariate sequences generated by agents, mining them for frequent patterns and using them for modeling. An example to this type of application is presented by Kaminka et al. [21] where the RoboCup soccer simulation games are used to model the various team behaviors.

One of the prominent applications of frequent sequence mining is classification of a sequential dataset. In [20], the behavior of people in a predefined group is observed, and frequent patterns are mined. These patterns are used in for classifying a given behavior as belonging to a specific person. The same application has been applied to a commercial setting in the personalized television domain. The work we will present is an extension of the mining component described in [20].

Support based algorithms for frequent sequence mining were first introduced by Agrawal and Srikant [2], where the algorithms AprioriAll, AprioriSome and DynamicSome were introduced. These algorithms naturally expand frequent itemset mining to frequent sequence mining. Itemsets do not contain a sequential ordering, whereas sequences do. The algorithms perform pattern mining in sequences of itemsets (events) and find frequent patterns in the input. The itemsets typically contain multiple items. Later they introduced the more efficient GSP [22] which has been broadly implemented and used since.

Since the search space for these mining problems is incredibly large other support based algorithms were introduced to improve the speed and efficiency of the mining process. SPADE [19], introduced by Zaki, is an algorithm for frequent sequence mining that belongs to the family of support based mining algorithms. SPADE outperforms GSP, due to the use of a vertical layout for the database and a lattice-theoretic approach for search space decomposition. We adopt the method presented in SPADE [19] and adapt it to use a normalized support for finding frequent sequences.

The key idea in many of these support based algorithms is the generation of candidate sequences. The candidate sequences are subsequences of the input-sequences in the database. *Frequent* candidate sequences are both placed in the set of mined *frequent* sequences, as well as used to generate the next generation of candidates. First, 2-sequences (sequences of length 2) are generated, then they are used to create 3-sequences etc: pairs of l-sequences with common prefixes are combined, to create an $l+1$-sequence.

Generating all possible candidate sequences is infeasible and results in an unscalable solution. Therefore, a pruning is introduced to this process. Candidates that are *not frequent* are pruned. They are not used to generate the next generation of candidates. The reason this can be done is based on the anti-monotonic property of support. Support has a nice anti-monotonic property promising that it does not grow when a candidate sequence is expanded. This promises that candidate sequences that are *not frequent* will never generate *frequent* sequences, and therefore can be pruned. Thus, the anti-monotonic property is very important and ensures scalability of the mining.

Alongside the rich variety of support based mining algorithms Mannila et al. [3] proposed an algorithm for mining frequent episodes, a type of frequent sequence, in an input composed of a single long sequence. Frequent *episode* mining algorithms find frequent items that are frequent within a single sequence whereas frequent support based sequence mining searches for items that reoccur in multiple sequences. Tatti and Cule [16] proposed mining closed episodes that are represented as DAGs. This algorithm cannot handle multivariate sequences. Salam and Khayal [23] introduced a method for mining top-k frequent patterns without the use of a minimum support. They generate patterns of length 2, and then use a top-down mechanism that only generates the top maximal frequent itemsets. They build a graphical representation of the data and search for maximal cycles in the graph.

The problem of Frequent Sequence Mining has been solved with many algorithms, an extended survey can be found in [24]. Often, the *frequent* sequences found are often insufficient. Unfortunately, support based mining methods suffer from a bias towards shorter sequences as has been shown in [10]. This means that in the frequent sequence mining, short sequences are found more often than long sequences. This is very problematic since these short sequences are often not very interesting as we will illustrate in Section V-E.

Several attempts have been made to address this bias. One possibility is to force large patterns by searching for closed patterns as in TSP [14] or maximal patterns such as MSPS [15]. However, mining closed or maximal patterns may not be the best approach to solve the short sequence bias. Using closed and maximal sequences ignores shorter partial sequences that may be of interest. We propose an algorithm that mines sequences of all lengths without a bias towards long or short sequences.

In LPMiner [25] (itemset mining) and SLPMiner [13] (sequence mining) Seno and Karypis introduce a length decreasing support constraint in order to overcome the short sequence bias. This is based on the observance that an interesting short sequence must be very frequent (have a very high support) to be interesting. Long sequences on the other hand may be interesting with a lower support. SLPMiner is a heuristic approach whereas in our work we attempt to find a general solution based on support normalization.

An alternative approach is taken by Yun and Legget in WSpan [12]. They introduce a weighted mining algorithm, for sequences with weighted items. Using weights in the mining process is very useful since it provides more input than using frequency alone. Unfortunately, this is of no assistance in domains where there is no information on what weights to apply. Our solution requires no knowledge on what weights should be used and can be implemented in any domain.

The methods for solving the bias towards short subsequences suggested in [12], [13], [25] are heuristic. They are based on forcing long sequences to be mined. In contrast, Horman and Kaminka [10] proposed using a statistical normalization of support. The support measure is normalized in relation to sequence length. They showed how support normalization enables finding frequent subsequences with different lengths in an unbiased fashion. Using normalized support makes no assumptions on the relation between length to support, or on the relative weights of the items in a database as were made in the other methods.

Normalization for frequent pattern mining has been performed in the past. SEARCHPATTOOL [4] uses z-score for normalization of mined frequent patterns. It first performs the sequence mining using a support based algorithm and then selects the significant sequences using the z-score measure. In [26] z-score is used to normalize the data in preprocessing stage, before any mining is performed. We also use the z-score measure for normalization. We show how to use the z-score measure for scale-up in the mining task. To the best of our knowledge this application of z-score is novel, and has been applied only in [10].

Although Horman and Kaminka [10] successfully solve the statistical bias using normalization, their method suffers from three problems. The first difficulty with the method proposed by Horman and Kaminka, solved in this paper involves the scalability of the algorithm. Using the normalized support ruins the anti-monotonic property used for pruning in support based mining. Unfortunately, this makes pruning impossible and therefore the algorithm is unscalable. The second difference between this paper their work is that as opposed to [10] where the mined sequences must be continuous in the original sequence, we allow holes in the sequence. An example would be if the original sequence is ABCD, the previous method can find the subsequences AB, ABC, ABCD, BC etc, but cannot mine ACD or ABD, whereas our algorithm mines both types. The third difference is that the previous method could not handle multiple attributes, as opposed to our approach that can.

With the scalability spoiled it seems there is a need to choose between a scalable algorithm to one that can fully overcome the short sequence bias. In this paper, we propose an algorithm that can do both. The algorithm we present uses normalized support to overcome the short sequence bias successfully while using a pruning method with a sampling unit to solve scalability issues.

III. Norm-Frequent Sequence Mining

Norm-Frequent Sequence Mining solves the short sequence bias present in traditional *Frequent* Sequence Mining. We begin by introducing the notation and the traditional *Frequent* Sequence Mining problem in Section III-A. We then define the *Norm-Frequent* Sequence Mining problem in Section III-B. We explain why the scalability is hindered by the naive implementation of normalized support and how this is resolved in Section III-C. Section III-C addresses scalability by introducing a bound that enables pruning in the candidate generation process and Section III-D describes the Sampling component. Finally, in Section IV, we bring all parts together to compose the REEF algorithm.

A. Notation and Frequent Sequence Mining

We use the following notation in discussing Norm Frequent Sequence Mining.

event Let $I = \{I_1, I_2, ..., I_m\}$ be the set of all *items*. An *event* (also called an *itemset*) is a non-empty unordered set of *items* denoted as $e = \{i_1, ..., i_n\}$, where $i_j \in I$ is an item. Without loss of generality we assume they are sorted lexicographically. For example, $e = \{ABC\}$ is an event with items A B and C.

sequence A *sequence* is an ordered list of *events*, with a temporal ordering. The sequence $s = e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_q$ is composed of q events. If event e_i occurs before event e_j, we denote it as $e_i < e_j$. e_i and e_j do not have to be consecutive events and no two *events* can occur at the same time. For example, in the sequence s=$\{ABC\} \rightarrow \{AE\}$ we may say that $\{ABC\} < \{AE\}$ since $\{ABC\}$ occurs before $\{AE\}$.

sequence size and length The *size* of a sequence is the number of events in a sequence, $size(\{ABC\} \rightarrow \{ABD\}) = 2$. The *length* of a sequence is the number of items in a sequence including repeating items. A sequence with length l is called an *l-sequence*. $length(\{ABC\} \rightarrow \{ABD\}) = 6$.

subsequence and contain A sequence s_i is a *subsequence* of the sequence s_j, denoted $s_i \preceq s_j$, if $\forall e_k, e_l \in s_i, \exists e_m, e_n \in s_j$ such that $e_k \subseteq e_m$ and $e_l \subseteq e_n$ and if $e_k < e_l$ then $e_m < e_n$. We say that s_j *contains* s_i if $s_i \preceq s_j$. E.g., $\{AB\} \rightarrow \{DF\} \preceq \{ABC\} \rightarrow \{BF\} \rightarrow \{DEF\}$.

database The database D used for sequence mining is composed of a collection of sequences.

support The *support* of a sequence s in database *D* is the proportion of sequences in *D* that *contain* s. This is denoted $supp(s, D)$.

This notation allows the description of multivariate sequence problems. The data is sequential in that it is composed of ordered events. The ordering is kept within the subsequences as well. The multivariate property is achieved by events being composed of several items. The notation enables discussion of mining sequences with gaps both in events and in items, as long as the ordering is conserved. The mined sequences are sometimes called patterns.

In traditional support based mining, a user specified minimum support called *minsup* is used to define frequency. A *frequent* sequence is defined as a sequence with a support higher than *minsup*, formally defined as follows:

Definition 1 (Frequent): Given a database D, a sequence s and a minimum support *minsup*. s is *frequent* if $supp(s, D) \geq minsup$.

The problem of frequent sequence mining is described as searching for all the *frequent* sequences in a given database. The formal definition is:

Definition 2 (Frequent Sequence Mining): Given a database D, and a minimum support *minsup*, find all the *frequent* sequences.

In many support based algorithms such as SPADE [19], the mining is performed by generating candidate sequences and evaluating whether they are frequent. In order to obtain a scalable algorithm a pruning is used in the generation process. The pruning is based on the anti-monotonic property of support. This property ensures that support does not grow when expanding a sequence, e.g., $supp(\{AB\} \rightarrow \{C\}) \geq supp(\{AB\} \rightarrow \{CD\})$. This promises that candidate sequences that are *not frequent* will never generate *frequent* sequences, and therefore can be pruned. *Frequent* sequence mining seems to be a solved problem with a scalable algorithm. However, it suffers from a bias towards mining short subsequences. We provide an algorithm that enables mining subsequences of all lengths.

B. Norm-Frequent Sequence Mining using Z-Score

In this section, we define the problem of *Norm-Frequent* Sequence Mining. We use the statistical z-score for normalization. The z-score for a sequence of length l is defined as follows:

Definition 3 (Z-score): Given a database D and a sequence s. Let $l = len(s)$ be the length of the sequence s. Let μ_l and σ_l be the average support and standard deviation of support for sequences of length l in D. The *z-score* of s denoted $\zeta(s)$ is given by $\zeta(s) = \frac{supp(s) - \mu_l}{\sigma_l}$.

We use the z-score because it normalizes the support measure relative to the sequence length. Traditional mining, where support is used to define frequency, mines sequences that appear often relative to **all** other sequences. This results in short sequences since short sequences always appear more often than long ones. Using the z-score normalization of support for mining finds sequences that are frequent relative to other **sequences of the same length**. This provides an even chance for sequences of all lengths to be found frequent.

Based on the definition of z-score for a sequence we define a sequence as being *Norm-Frequent* if the z-score of the sequence is among the top z-score values for sequences in the database. The formal definition follows:

Definition 4 (Norm-Frequent): Given a database D, a sequence s of length l and an integer k. Let Z be the set of the k highest z-score values for sequences in D, s is *norm-frequent* if $\zeta(s) \in Z$. In other words, we perform top-K mining of the most norm-frequent sequences.

We introduce the problem of *Norm-Frequent* Sequence Mining. This new problem is defined as searching for all the *norm-frequent* sequences in a given database. The formal definition follows and will be addressed in this paper.

Definition 5 (Norm-Frequent Sequence Mining): Given a database D and integer k, find all the *norm-frequent* sequences.

In Figure 1, we provide a small example. The sequences $\{AB\}$, $\{A\} \rightarrow \{A\}$ and $\{B\} \rightarrow \{A\}$, of length 2, all have a support of 0.4 and are the most frequent patterns using support to define frequency. Notice that there are several sequences with this support, and no single sequence stands out. Consider the sequence $\{AB\} \rightarrow \{A\}$ of length 3. This sequence only has a support of 0.3. However, all other sequences of length 3 have a support no higher than 0.1. Although there are several sequences of length 2 with a higher support than $\{AB\} \rightarrow \{A\}$, this sequence is clearly interesting when compared to other sequences of the same length. This example provides motivation for why support may not be a sufficient measure to use. The norm-frequency measure we defined is aimed at finding this type of sequence.

Unfortunately, the z-score normalization test hinders the anti-monotonic property: we **cannot** determine that $\zeta(\{AB\} \rightarrow \{C\}) \geq \zeta(\{AB\} \rightarrow \{CD\})$.
Therefore, pruning becomes difficult; we cannot be sure that the z-score of a candidate sequence with length l will not improve in extensions of length $l + 1$ or in general $l + n$

seq 1:	$\{AB\} \rightarrow \{A\}$
seq 2:	$\{AB\} \rightarrow \{B\}$
seq 3:	$\{BC\} \rightarrow \{A\}$
seq 4:	$\{AB\} \rightarrow \{A\}$
seq 5:	$\{BC\} \rightarrow \{B\}$
seq 6:	$\{AC\} \rightarrow \{B\}$
seq 7:	$\{AB\} \rightarrow \{A\}$
seq 8:	$\{AC\} \rightarrow \{C\}$
seq 9:	$\{BC\} \rightarrow \{C\}$
seq 10:	$\{AC\} \rightarrow \{A\}$

Figure 1: Example database.

for some positive n. Therefore, we cannot prune based on z-score and ensure finding all *norm-frequent* sequences. This is a problem since without pruning our search space becomes unscalable.

Another problem with performing *Norm-Frequent* Sequence Mining is that the values for μ_l and σ_l must be obtained for sequences of all lengths prior to the mining process. This imposes multiple passes over the database and hinders scalability.

These important scalability issues are addressed and solved in Section III-C resulting in a scalable frequent sequence mining algorithm that overcomes the short sequence bias.

C. Scaling Up

As we explained in Section III-B, pruning methods such as those described in SPADE [19] cannot be used with *norm-frequent* mining. We propose an innovative solution that solves the scalability problem caused by the inability to prune.

Our solution is to calculate a bound on the z-score of sequences that can be expanded from a given sequence. This bound on the z-score of future expansions of candidate sequences is used for pruning. We define the bound and then explain how it is used. Z-score was defined in Definition 3. The bound on z-score is defined in Definition 6.

Definition 6 (Z-score-Bound): Given a database D and a sequence s. Let $\mu_{l'}$ and $\sigma_{l'}$ be the average support and standard deviation of support for sequences of length l' in D. The *z-score-bound* of s, for length l' denoted $\zeta^B(s,l')$ is given by $\zeta^B(s,l') = \frac{supp(s)-\mu_{l'}}{\sigma_{l'}}$.

We know that support is anti-monotonic, therefore, as the sequence length grows support can only get smaller. Given a candidate sequence s of length l with a support of $supp(s)$ we know that for all sequences s' generated from s with length $l' > l$ the maximal support is $supp(s)$. We can calculate the bound on z-score, $\zeta^B(s,l')$, for all possible extensions of a candidate sequence. Notice that for all sequences s' that are extensions of s, $\zeta(s') \leq \zeta^B(s,l')$. The ability to calculate this bound on possible candidate extensions is the basis for the pruning.

In order to mine *frequent* or *norm-frequent* sequences, candidate sequences are generated and evaluated. In traditional *frequent* sequence mining there is only one evaluation performed on each sequence. If the sequence is found to be *frequent* it is both saved in the list of *frequent* sequences and expanded to generate future candidates, if it is not *frequent* it can be pruned (not saved and not used for generating candidates). For *norm-frequent* mining we perform two evaluations for each sequence. The first is to decide whether the proposed sequence is *norm-frequent*. The second is to determine if it should be expanded to generate more candidate sequences for evaluation. There are two tasks since z-score is not anti-monotonic and a sequence that is not *norm-frequent* may be used to generate *norm-frequent* sequences. This second task is where the bound is used for pruning. The bound on future expansions of the sequences is calculated for all possible lengths. If the bound on the z-score for all possible lengths is lower than the top n z-scores then no possible expansion can ever be *norm-frequent* and the sequence can be safely pruned from the generation process. If for one or more lengths the bound is high enough to be *norm-frequent* we must generate candidates from the sequence and evaluate them in order to determine if they are *norm-frequent* or not. This process guarantees that all *norm-frequent* sequences will be generated.

Using the bound enables pruning of sequences that are guaranteed not to generate *norm-frequent* candidates. The pruning enabled by using the bound resolves the first scalability issue of sequence pruning in the generation process. The second scalability problem of calculating μ_l and σ_l is resolved by calculating the values for μ_l and σ_l on a small sample of the data in a preprocessing stage described below.

D. Sampling for Norm-Frequent Mining

Norm-frequent mining uses the z-score defined in Definition 3 and the bound described in Definition 6. Both these measures make use of the average and standard deviation of support for each subsequence length (μ_l and σ_l). We must calculate these values prior to the sequence mining. The naive way to calculate these values would be to generate all possible subsequences and calculate these measures. However, this is obviously irrelevant as making a full expansion completely defeats the purpose of mining with the z-score pruning.

Therefore, we propose extracting a small sample of the database and calculating these values on the sample. For the sample, full expansion is feasible and generates the necessary measures while ensuring scalability.

However, there is a problem that arises with the sampled measures. They do not reflect the full database measures correctly. It has been shown by [11], [15], [27], [28], that there is a distortion, also termed overestimation, in the values of support calculated on a sample of a database relative to support calculated over a full database. Similarly, the average and standard deviation of support suffer a distortion in the sampled data.

1) Effects of Sampling Distortion: We use Figure 2 to demonstrate how the distortion affects sequence mining. The norm-freq sequences that are mined using z-score and calculated with statistics from the full database are displayed in column 1. The top most stripes (light) represent the most *norm-frequent* sequences and the bottom (dark) represent sequences that are *not norm-frequent* (rare). Column 3 shows the *norm-frequent* sequences discovered using averages and standard deviations for z-score calculation from the sampled database, the colors match the coloring in the full database, the ordering is based on sampled results. One notices that the order is confused and rare sequences in dark greys show up relatively high in the list. *Norm-frequent* (light) sequences are pushed down as rare. The black stripes at the side of the column represent sequences that did not appear at all in the *norm-frequent* list when using the full data set and appeared when using the sampling. It is obvious that sequences are shifting around and *norm-frequent* sequences are being chosen as rare and vice versa. Therefore, the distortion badly affects sequence mining. In column 2 we use the correction displayed in the next section and improve this shifting. The rare sequences show up further down with the correction than without, as do the candidates that did not appear in the original list. Although this is just an example on one small set of data it conveys the effects of the distortion and the correction.

2) Chernoff Bounds and Hoeffding Inequalities: We would like to evaluate how far off the sampled statistics are from the real statistics. One might suggest using Chernoff bounds as in [27], [29] or Hoeffding inequalities as in [28] for this task. In [27]–[29], the aim is to show how far off sampled support is from real support for a single subsequence. The appearance of a subsequence in each sequence in the sample is described as a random variable with a Bernoulli distribution. These random variables are **independent**, and the Chernoff bounds or Hoeffding inequalities can be used. The scenario we are using is different. Instead of looking at the accuracy of the support on the sampled data we are looking at the **average** and **standard deviation** of support for a subsequence of a specific length. Unlike the application of Chernoff and Hoeffding bounds in [27]–[29], where the random variable was independent, the random variable in our setting is the average of support of sequences for a given length. This random variable is strongly **dependent** and therefore the known bounds are problematic to apply. There are also situations where although Chernoff bounds can be applied, it is problematic to apply them because a very large sample of the database is needed, as in [27]. For cases where Chernoff and Hoeffding bounds cannot be applied, or situations where one chooses not to apply them we propose a method of distortion correction.

3) Sampling Distortion Correction Method: We introduce a method for correcting the distortion that can be used for any data set. This method finds the model of the distortion for various input sequence lengths and sample rates using the non-linear regression function *nlsfit()* in the R Project for Statistical Computing [30]. Once we have modeled the distortion, correcting it is immediate. The model provides an equation that determines the exact distortion value of average support or standard deviation for a given input sequence length and sample rate. A simple inverse multiplication provides the corrected value.

In order to perform the regression we must propose functions and then perform the non-linear regression to set the parameters. We list the equations we propose using based on our experimental experience as described in Section V-B.

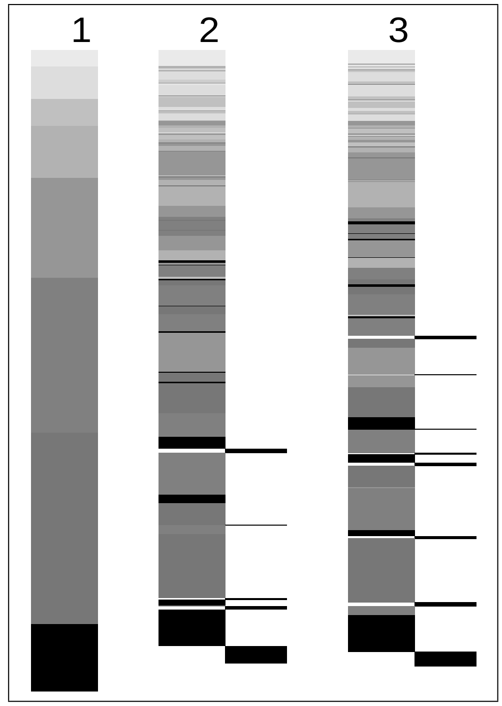


Figure 2: Sampling distortion effect.

The variables in the equations are len (length of the input-sequences) and smp (sampling rate). The coefficients that are determined in the non-linear regression are a, b, c, d. The functions we propose using are:

$$a \times (len - 1)^{b \times log(smp)} + c \times smp \quad (1)$$

$$a \times (len)^{b \times smp + c} + d \times smp \quad (2)$$

$$a \times (len - 1)^{b \times smp + c} + d \times smp \quad (3)$$

$$(len - 1)^{a} + e \times smp^{b} + c \times len + d \times smp \quad (4)$$

For the standard deviation of support we perform distortion correction in a similar fashion using the following equations for approximation.

$$a * (smp^{b}) \quad (5)$$

$$(smp^{b}) + a \quad (6)$$

A tool such as the R Project for Statistical Computing [30] is used to find the correct parameters for these equations as demonstrated in detail in Section V-B.

Once we have found the equations that represent the distortion for average support and standard deviation of support, for a certain type of data set, correction of this distortion is simple. For new data in these sets we can select any sample rate and calculate the distortion correction of average and standard deviation for each possible sequence length. We multiply the sampled values by the inverse of the distortion and use the results as the average and standard deviation of support in the z-score calculation for *norm-frequent* mining.

We found the proposed equations to be general and provide good approximations for different data sets, as shown in

Section V-B, and therefore suggest they can be used for other data sets as well. However, for data sets where these equations do not provide good approximations the same method we used can be applied while using different equations.

Now that we have presented the full method for sampling the data set and calculating the values for μ_l and σ_l we have a complete scalable algorithm that mines *norm-frequent* sequences without a bias to short sequences, Section IV puts all the pieces together and describes the full algorithm.

IV. REEF ALGORITHM

In this section, we combine all the components we have described in the previous sections and describe the implementation of REEF. The REEF algorithm is composed of several phases. The input to REEF is a database of sequences and an integer *'k'* determining how many Z-scores will be used to find *norm-frequent* sequences. The output of REEF is a set of *norm-frequent* sequences. Initially, a sampling phase is performed to obtain input for the later phases. Next we perform the candidate generation phase. First, norm-frequent 1-sequences and 2-sequences are generated. Once 2-sequences have been generated, an iterative process of generating candidate sequences is performed. The generated sequences are evaluated, and if found to be *norm-frequent* are placed in the output list of *norm-frequent* sequences. These sequences are also examined in the pruning process of REEF in order to determine if they should be expanded or not.

Sampling Phase - The sampling phase is performed as a preprocessing of the data in order to gather statistics of the average and standard deviation of support for sequences of all possible lengths. This stage uses SPADE [19] with a *minsup* of 0 to enumerate all possible sequences in the sampled data and calculate their support. For each length the support average and standard deviation are calculated. These values are distorted and corrected values are calculated using the technique described in Section III-D. These corrected values provide the average support μ_l and standard deviation of support σ_l that are used in z-score calculation and the bound calculation.

Candidate Generation Phase - The candidate generation phase is based on SPADE along with important modifications. As in SPADE we first find all 1-sequence and 2-sequence candidates. The next stage of the candidate generation phase involves enumerating candidates and evaluating their frequency.

We make two modifications to SPADE. The first is moving from setting a *minsup* to setting the $'k'$ value. $'k'$ determines the number of z-score values that norm-frequent sequences may have. Note that there may be several sequences with the same z-score value. The reason for this modification is that z-score values are meaningful for comparison within the same database but vary between databases. Therefore, setting the $'k'$ value is of more significance than setting a min-z-score threshold.

The second and major change we make is swapping *frequency* evaluation with *norm-frequency* evaluation. In other words, for each sequence s replace the test of is $supp(s, D) > minsup$ with the test of is $\zeta(s) \in Z$ where Z is the set of the $'k'$ highest z-score values for sequences in D. This replacement of the frequency test with the norm-frequency test is the essence of REEF and our main contribution.

1: **for all** x is a prefix in S **do**
2: $T_x = \emptyset$
3: $F_R = \{$k empty sequences$\}$
4: **for all** items $A_i \in S$ **do**
5: **for all** items $A_j \in S$, with $j \geq i$ **do**
6: $R = A_i \bigvee A_j$ (join A_i with A_j)
7: **for all** $r \in R$ **do**
8: **if** $\zeta(r) > \zeta($a seq s in $F_R)$ **then**
9: $F_R = F_R \bigcup r \backslash s$ //replace s with r
10: **for all** $l' = l+1$ to *input sequence length* **do**
11: **if** $\zeta^B(r, l') > \zeta($a seq s in $F_R)$ **then**
12: **if** A_i appears before A_j **then**
13: $T_i = T_i \bigcup r$
14: **else**
15: $T_j = T_j \bigcup r$
16: enumerate-Frequent-Seq-Z-score(T_i)
17: $T_i = \emptyset$

Figure 3: Enumerate-Frequent-Seq-Z-score(S). Where S is the set of input sequences we are mining for frequent subsequences, A set of *norm-frequent* subsequences is returned, F_R is a list of sequences with the top $'k'$ z-scores.

The improved version of sequence enumeration including the pruning is presented in Figure 3 and replaces the enumeration made in SPADE. The joining of *l*-sequences to generate *l+1*-sequences ($A_i \bigvee A_j$ found in line 6) is performed as in SPADE [19].

Pruning Phase using Bound - Obviously REEF cannot enumerate all possible sequences for norm-frequency evaluation. Furthermore, as we discussed in Section III-B, the z-score measure is not anti-monotonic and cannot be used for pruning while ensuring that norm-frequent candidates are not lost. In Section III-C, we introduced the bound on z-score that is used for pruning.

The pruning in REEF calculates $\zeta^B(s, l')$ for all possible lengths $l' > l$ of sequences than could be generated from s. The key to this process that there is no need to actually generate the extensions s' that can be generated from s. It is enough to know the $supp(s)$, μ_l and σ_l for all $l' > l$. If for any length $l' > l$ we find that $\zeta^B(s, l') \in Z$ (in the list of 'k' z-scores) we keep this sequence for candidate generation, if not then we prune it. Using the bound for pruning reduces the search space while ensuring closure or in other words ensuring all frequent sequences are found. The pruning is performed as part of the enumeration described in algorithm Figure 3. This pruning is the key to providing a **scalable** *norm-frequent* algorithm.

V. Evaluation

In this section, we present an evaluation of REEF on several data sets, described in Section V-A. We first demonstrate how to use our sampling distortion method in Section V-B. Next, in Section V-C we compare runtime of the algorithms and justify the use of the bound that was introduced in Section IV. Then in Section V-D will show that *norm-frequent* mining overcomes the short sequence bias present in *frequent* mining algorithms. In Section V-E, we will provide evidence that the sequences mined with REEF are more meaningful than sequences mined with SPADE.

A. Data Sets and Experimental Settings

The evaluation is performed on 4 data sets. One of these is a synthetic data set, three use real world sequential data.

Syn is the synthetic data generated with the IBM QUEST Synthetic Data Generator [31]. QUEST generates data for various data mining tasks, including frequent sequence mining. We generated sequences with the following parameters: Number of customers in database = 1000, Average number of intervals per sequence = 3, Average number of transactions per interval = 3, Number of items = 10, all other settings are the default settings. The tests in the evaluation are performed on 5 synthetic sets with these parameters.

TEXT is a corpus of literature of various types. We treat the words as sequences with letters as single item events. We removed all formatting and punctuation from text (apart from space characters) resulting in a long sequence of letters. Mining this sequential data for frequent sequences produces sequences of letters that may or may not be real words. The reason we chose to mine text in this fashion is to show how interesting the frequent sequences are in comparison to norm-frequent sequences by testing how many real words are discovered. In other words, we use real words from the text as ground truth against which to evaluate the algorithms. We use three sets of textual data, one is from Lewis Carroll's "Alice's Adventures in Wonderland" [32], another is Shakespeare's "A Midsummer Night's Dream" [33] and the third is a Linux installation guide [34]. Evaluation is performed on segments of the corpus. Each test is performed on five segments.

UPD: **U**ser **P**attern **D**etection, is a data set composed of real world data used for evaluation. UPD logs keyboard and mouse activity of users on a computer as sequences. Sequences mined from the UPD data can be used to model specific users and applied to security systems as in [35], [36] and [20]. The experiments are run on 11 user sessions. The data is collected throughout the whole work session and not just at login. Each activity is logged along with the time and date it occurs. The data is then converted into the following events: pressing a key, time between key presses, key-name, mouse click, mouse double click, time between mouse movements. For each session the events are saved in sequences.

Zapping is composed of data that we gathered on remote control usage. In each household members were asked to identify themselves as they begin watching TV, by pressing a designated button on the remote, and then the "zapping sequence" is saved, in other words the buttons they pressed on the remote while they were watching. This sequence is converted into the following events: Button pressed, Time passed since last activity and Time of day. This interesting data set in the domain of personalized television learns personal usage patterns to provide personal services as in [37] and [20]. Each zapping session generates a single long sequence. Evaluation is performed on 10 sets.

For all these data sets the input is composed of long sequences. In order to use REEF these sequences are cut into smaller sequences using a sliding window thus creating manageable sequences for mining. The size of the sliding window is termed *input sequence length* in our results. The comparison made between REEF to SPADE is delicate since SPADE uses *minsup* to define how many sequences to mine whereas REEF uses *'best'* as described in Section IV. Adjusting these settings changes the runtime and may change the quality and lengths of the mined sequences. Although these parameters are similar in nature they cannot be set to be exactly the same for experiments. We consistently use a single setting of *minsup=1%* and *'k'=50* throughout all experiments and a sample rate of 10% for the preprocessing sampling component.

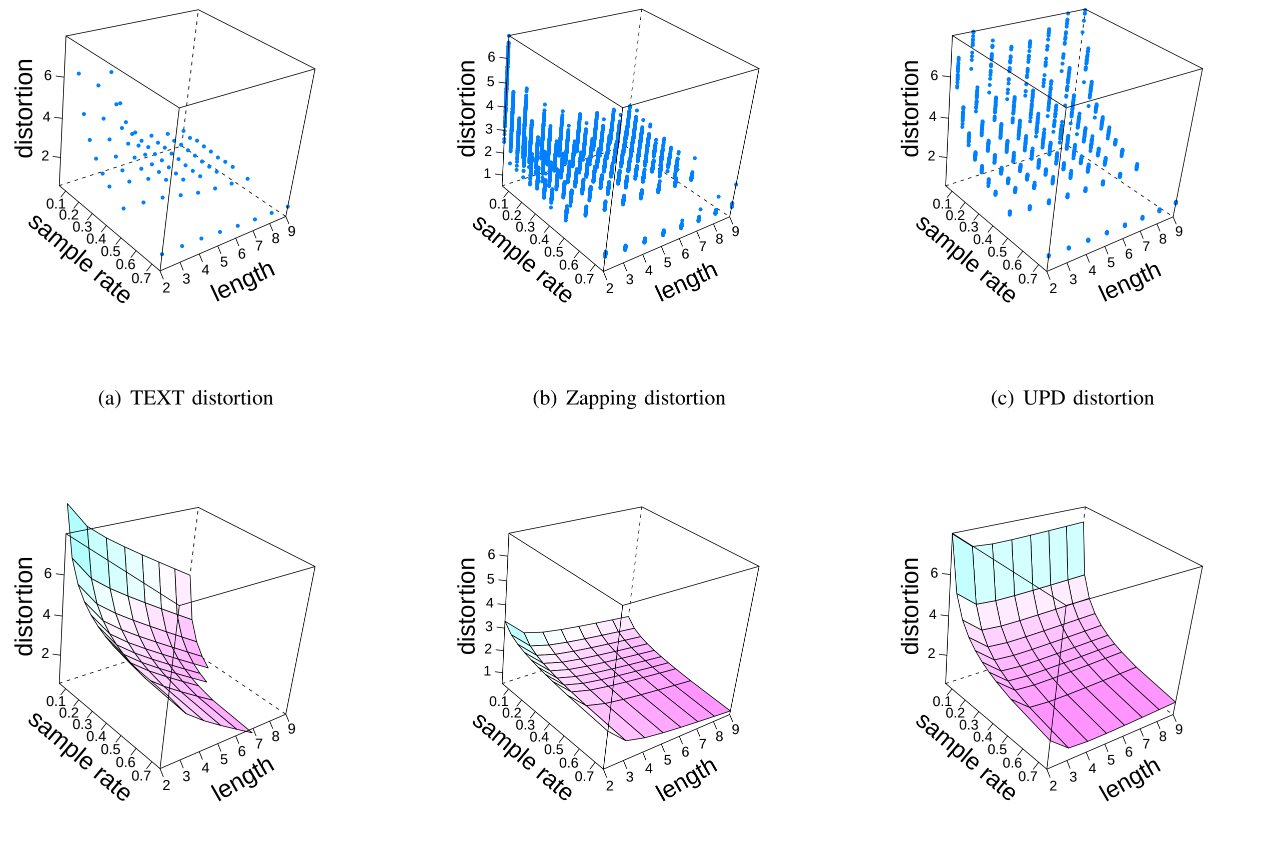
B. Sampling Distortion Correction

We will demonstrate how to perform the distortion correction described in Section III-D3, for several data sets. We found a single equation to model distortion for all the data sets we investigated. Although this does not imply that the same model fits all possible data sets, it is a strong indication that this may be the case. For data sets where this does not hold, the same method we used can be applied to find other models. The data sets we used are the TEXT data set the Zapping data set and the UPD dataset (described in the previous section).

Figure 4 (a),(b),(c) displays the distortion ratio between sampled average support to full data average support on all three data sets. The data used for this analysis is excluded from the experimental evaluation performed in Sections V-C, V-D and V-E. We used approximately half the data for this analysis and half for the experimental evaluation. Each point is an instance of the dataset. The distortion is calculated on each instance for various input sequence lengths and sample rates. The distortion obviously has an orderly structure that we want to find.

We modeled the distortion using non-linear regression. We used R Project for Statistical Computing [30] in order to find a general formula for calculating the correction factor. We need two correction parameters: one for the average support, the other for the standard deviation of support.

We first describe the average support correction. We noticed that when we set the sample rate, the distortion ratio follows a nonlinear function of the length, shown in Figure 5. On the other hand, if we set the length, then the distortion ratio follows a nonlinear function of the sample rate, shown in Figure 6. Therefore, the distortion of average support is dependent both on length and on sample rate and we are



(a) TEXT distortion (b) Zapping distortion (c) UPD distortion

(d) TEXT regression (e) Zapping regression (f) UPD regression

Figure 4: Distortion ratios of average support on sampled data in (a), (b) and(c). Regression surfaces of equation (4) in (d),(e) and (f).

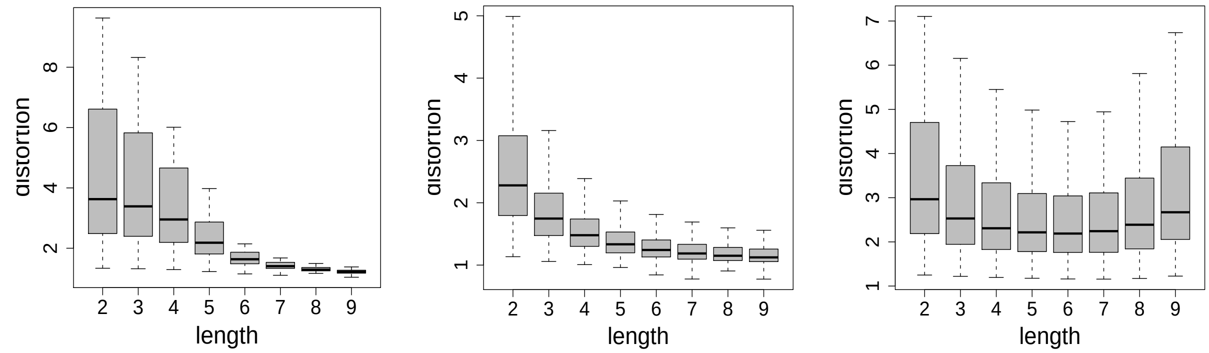
looking for a function $f(len, smp)$ where *len* is the length of a sequence and *smp* is the sample rate.

In previous research [20], we investigated the distortion on the Zapping data alone. We tried to build a combination of the power and logarithmic functions that we saw when looking at each variable, into a single function. This led us to investigating Equations (1), (2) and (3) in Section III-D3. However, when we tried performing regression for other data sets we discovered that for UPD these were not the best candidates, and did not even converge on the TEXT data. We suspect over-fitting of the regression on the Zapping data. Realizing that the shape of the distortion is reminiscent of a stretched paraboloid we tried regression with Equation (4) in Section III-D3 and found that this best suits all three data sets and was therefore selected as the distortion model. The regression surfaces for Equation (4) in Section III-D3 appear in Figure 4 (d),(e),(f). Values of the parameters for non linear least of squares regression appear in Table I.

Standard deviation of distortion is linear relative to length (see Figure 7), and is a nonlinear function of sample rate (see Figure 8). Therefore, the only variable involved is the sample rate. The sampling distortion correction we found for zapping in [20] fits the UPD and TEXT data as well. The equations we tested are Equations (5) and (6) in Section III-D3, we chose Equation (5). Regression parameters appear in Table II.

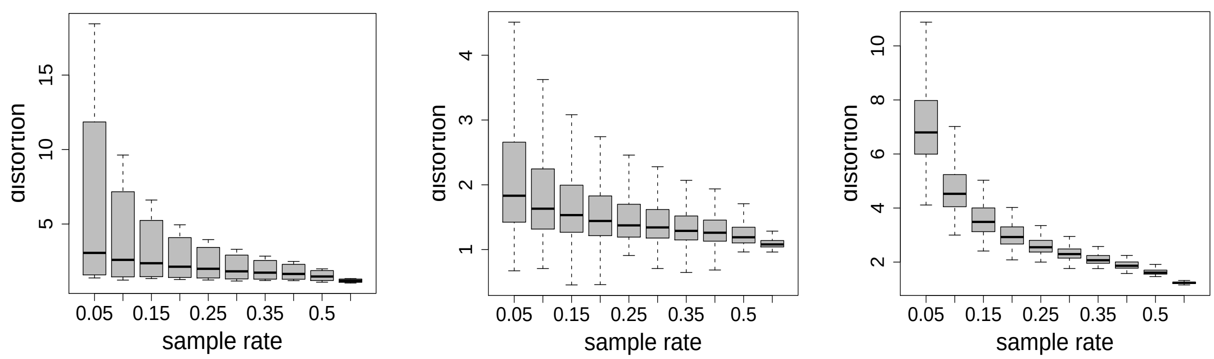
C. REEF Runtime and Bound Pruning

We show how the use of the bound enables speedup. Although the main focus of REEF is not speed but rather output quality, we show that REEFs' runtime is comparable with existing algorithms. We compare the runtime for two versions of REEF to SPADE. REEF refers to the full algorithm described in Section IV. NB-REEF refers to the same algorithm but without the use of the bound, or in other words without pruning. The runtime includes both the sampling time and the actual mining time for both REEF and NB-REEF. We compare REEF in two versions, one with the use of the bound and the other without. We added SPADE for completeness. We knew in advance that SPADE is faster than our algorithm, but displaying runtime for SPADE provides an order of magnitude for the bound comparison. Results for all data sets appear in Figure 9. There are four types of data sets as described in



(a) TEXT (b) Zapping (c) UPD

Figure 5: Length cross cut of distortion ratio for average support.



(a) TEXT (b) Zapping (c) UPD

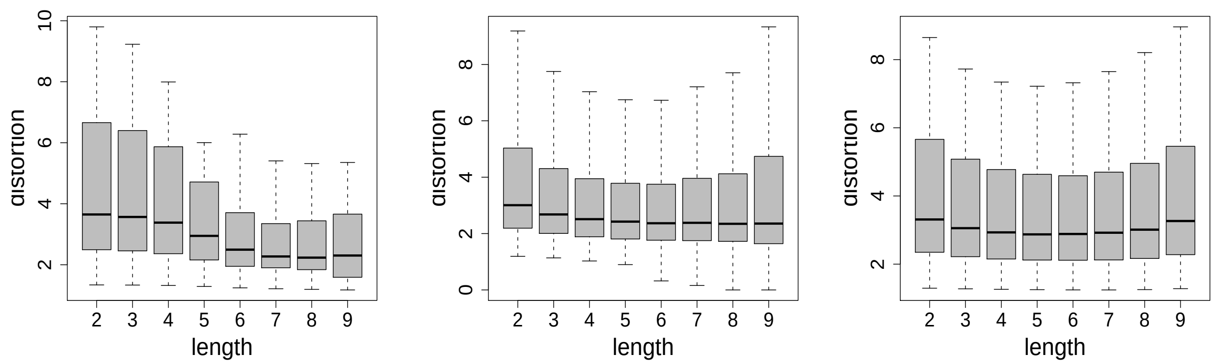
Figure 6: Sample rate cross cut of distortion ratio for average support.

TABLE I: Regression parameter values for average support.

data set	func	RSE	a	b	c	d	e
Zapping	1	0.3975	3.561935	0.183471	-3.438119		
Zapping	2	0.3596	5.109377	0.751144	-0.528501	-5.712356	
Zapping	3	0.3391	3.789649	0.703963	-0.465705	-4.124942	
Zapping	4	0.3998	-1.664660	-0.229604	-0.067215	-0.087310	1.226132
UPD	1	1.21	4.935761	-0.009716	-6.576244		
UPD	2	1.102	8.852121	0.648309	-0.302588	-15.3049	
UPD	3	1.141	6.932194	0.476740	-0.220498	-10.9230	
UPD	4	0.5916	-2.417082	-0.665367	0.037497	-0.392893	0.928968
TEXT	4	1.899	-1.416	-0.409	-0.559	1.387	2.719

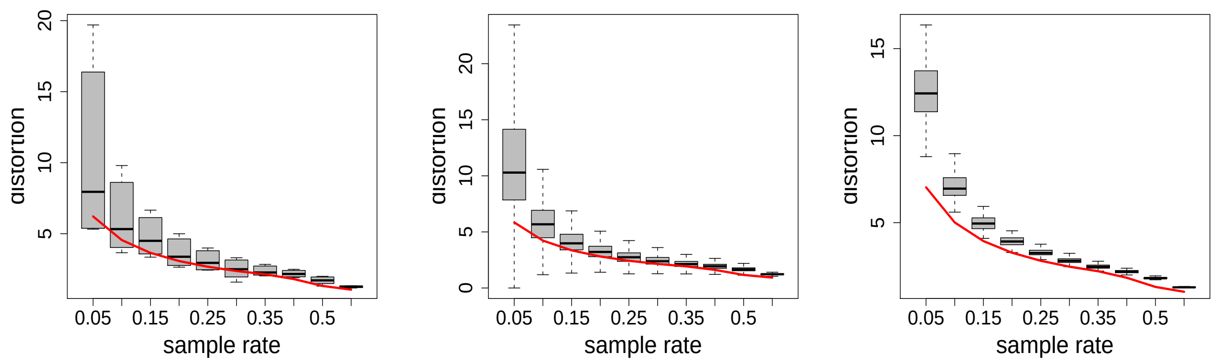
TABLE II: Regression parameter values for standard deviation of support.

num	func	RSE	a	b
Zapping	5	2	0.928601	-0.799307
Zapping	6	2	-0.101002	-0.776648
UPD	5	0.6055	1.027204	-0.835350
UPD	6	0.6057	0.043373	-0.843472
TEXT	5	2.053	1.059	-0.768



(a) TEXT (b) Zapping (c) UPD

Figure 7: Length cross cut of distortion ratio for standard deviation.



(a) TEXT (b) Zapping (c) UPD

Figure 8: Sample rate cross cut of distortion ratio for standard deviation.

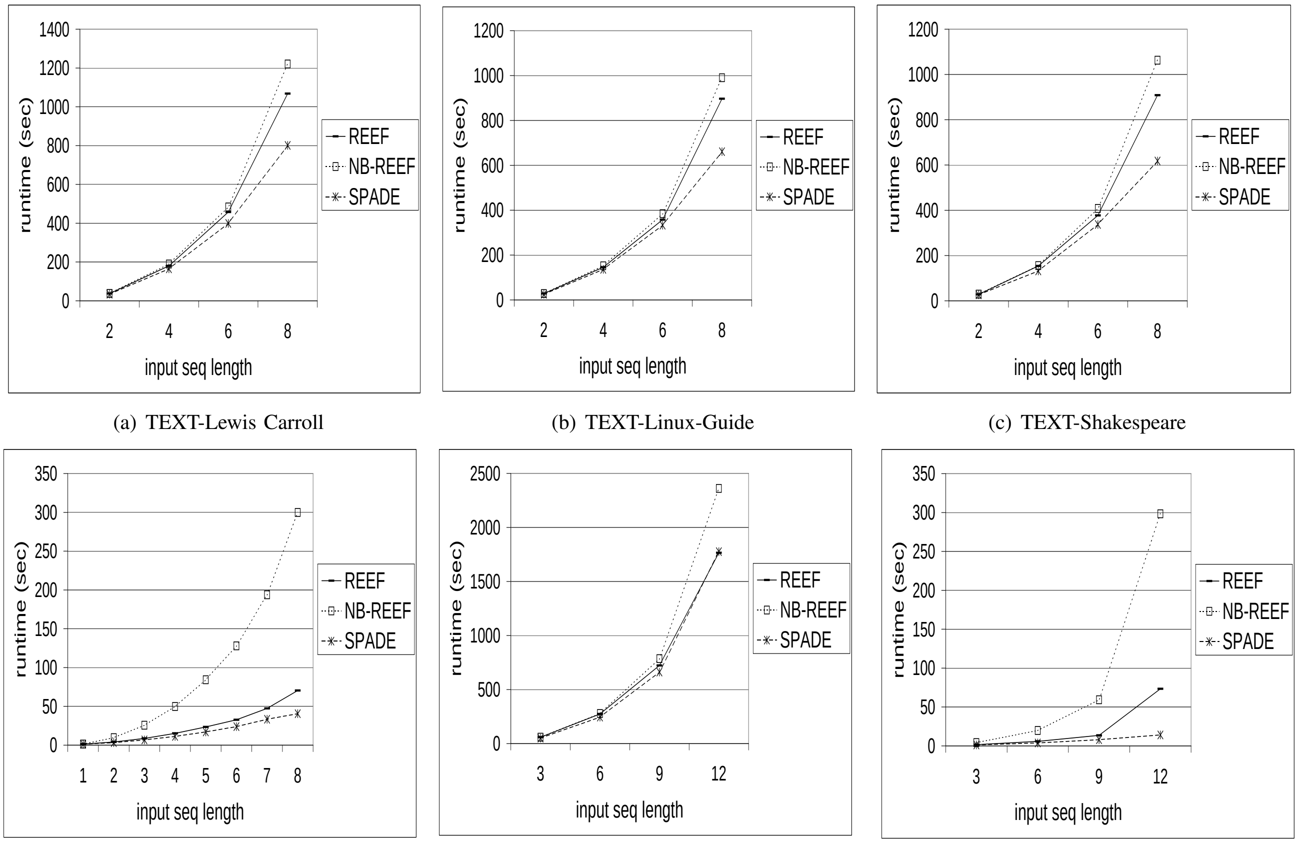
Section V-A, however the TEXT dataset is composed of three sets of textual data, thus in the results in Figure 9 there are 6 graphs. The x-axis represents input-sequence length. For the synthetic data we had full control over input sequence length and thus present results for all values. For the real data sets the input sequence length is controlled by the number of attributes in an event. This results in varying values along the x-axis for the results. The y-axis displays runtime of the algorithm in seconds. We tested the runtime for various input-sequence lengths. Each point on the graph is the average of five runs.

The first important observation to make is the importance of the pruning bound. For all data sets the pruning noticeably reduces runtime and is an important component of REEF. This is particularly noticeable on the synthetic data in Figure 9(d), UPD data in Figure 9(e) and in the Zapping data in Figure 9(f). This difference grows with input sequence length and becomes more important as input length grows.

The other important result is that the REEF runtime is comparable with that of SPADE. Although SPADE is faster than REEF they are close in runtime. The reason SPADE is often faster than REEF is because *minsup* provides a tighter pruning bound than the one we use in REEF. However, faster may not be better. The tight pruning results in the creation of short sequences. In the next section, we show that there is a tradeoff between runtime to the length of mined sequences, and show how REEF although slightly slower than SPADE has better performance. By overcoming the short sequence bias REEF produces a better distributed set of mined sequences.

D. Resolving Length Bias in Frequent Sequence Mining

In this section, we establish how REEF successfully overcomes the short sequence bias that is present in the frequent sequence mining techniques. We performed *frequent* sequence mining with SPADE and *norm-frequent* sequence mining with REEF. We compared the lengths of the mined sequences for both algorithms. The results are displayed in Figure 10. Results are shown for the Syn, UPD, Zapping and three TEXT data sets. The x-axis shows the lengths of the mined sequences. The y-axis displays the percentage of sequences found with



(a) TEXT-Lewis Carroll (b) TEXT-Linux-Guide (c) TEXT-Shakespeare

(d) Synthetic (e) UPD (f) Zapping

Figure 9: Runtime. Comparing REEF (with bound), NB-REEF (without bound) and SPADE.

the corresponding length. For each possible length we counted the percentage of mined sequences with this length.

The synthetic data set in Figure 10(d) displays the clearest description of the algorithmic behavior. While SPADE outputs mainly sequences with a length of 2, some with a length of 3, very little with a length of 4 and no longer sequences, REEF outputs sequences with lengths varying from 2 to 6 and with a bell shaped distribution. REEF captures the real nature of the synthetic data and the correct distribution of sequence length.

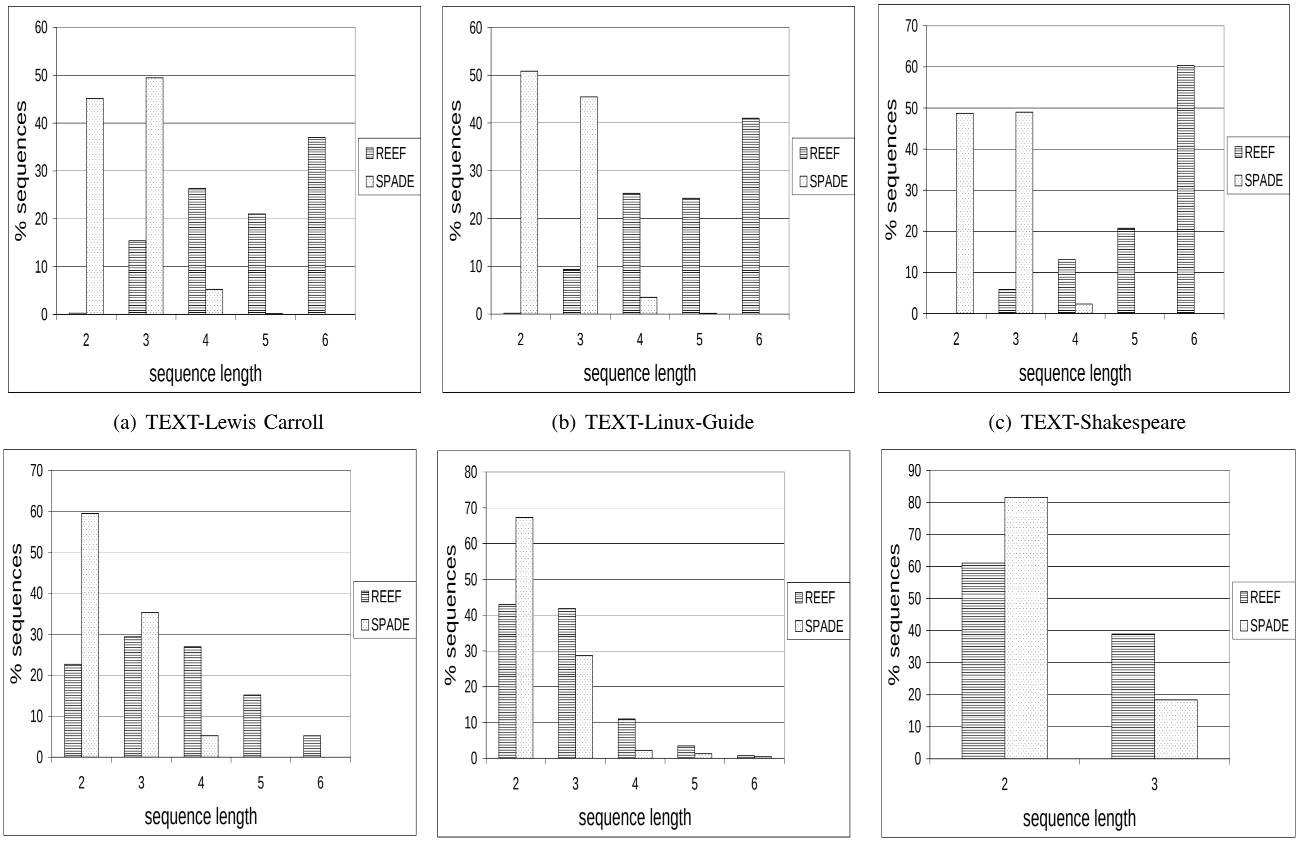
In the TEXT data set, the results on all three text corpora show how SPADE mines mainly short sequences, while REEF manages to mine a broader range of sequence lengths as displayed in Figure 10(a),(c),(b). REEF results are much closer to known relation between word length to frequency [38] than the SPADE output. In the next section, we count how many of these sequences are words to illustrate superiority of REEF.

For the Zapping and UPD data REEF again overcomes the short sequence bias and provides output sequences of all lengths in a more normal distribution than with SPADE. This can be seen in Figure 10(e). Note that in contrast to the TEXT corpora, there is no known ground truth as to what the length of frequent sequences should be in this domain, and what their distributions are. Thus, there is no way to confirm whether we have found the correct distribution. However, we do show that we are not restricted to mining short sequences.

An interesting data set is the Zapping set. Although REEF allows for fair mining of all lengths the sequences found both with REEF and with SPADE are short, and there are no sequences with lengths higher than 3 as shown in Figure 10(f). This seems to imply that the frequent sequences in this set really are short. For this data set it would be more beneficial to use SPADE than REEF since there is not much quality to be gained from the slightly longer runtime with REEF. The Zapping set is different to all other three sets, where the extra runtime is clearly worthwhile, since the output sequences tend to be better representatives of the data set. Results on all four sets clearly show the tradeoff in the mining algorithms between time to sequence quality. Frequent sequence mining in support based algorithms such as SPADE generate short frequent sequences quickly. In contrast, norm-frequent mining such as the one we presented in REEF takes slightly longer, but generates sequences with a broader length distribution as we show in Section V-E.

E. Mining Meaningful Sequences with REEF

The text domain was chosen specifically in order to illustrate the quality of the output sequences. We wanted a domain where the meaning of interesting sequences was clear. TEXT is obviously a good domain for this purpose since words are clearly more interesting than arbitrary sequences of letters.



(a) TEXT-Lewis Carroll (b) TEXT-Linux-Guide (c) TEXT-Shakespeare

(d) Synthetic (e) UPD (f) Zapping

Figure 10: Removal of length bias.

We hope to find more real words when mining text than nonsense words. Our evaluation is performed on three sets of text as described above. Results appear in Figure 11. We compare results on *frequent* sequence mining using SPADE with *norm-frequent* sequence mining using REEF. The x-axis shows different input sequence lengths (window sizes). For each input sequence length we calculated the percentage of real words that were found in the mined sequences. This is displayed on the y-axis. For example the top 15 mined sequences in Shakespeare using REEF: {*e he,or,e and,her,n th,though,he,s and,her,thee,this,thou,you,love,will*}and using SPADE: {*rth,mh,lr,sf,tin,op,w,fa,ct,ome,ra,yi,em,tes,t l*} Using REEF yields many more meaningful words than using SPADE.

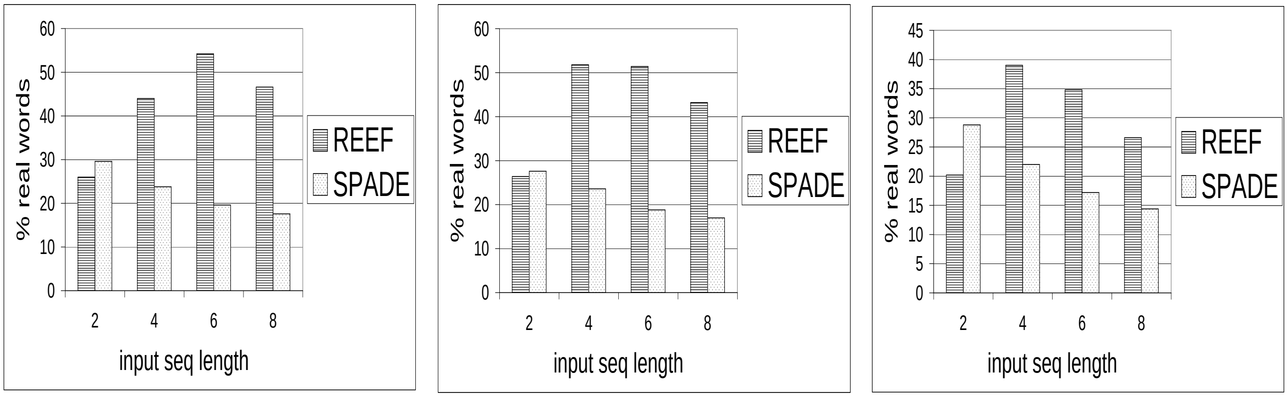
For all text sets REEF clearly outdoes SPADE by far. REEF manages to find substantially more words than SPADE for all input lengths. The short input-sequence sizes of 2 does not produce high percentages of real words for REEF or SPADE. Using longer input sequence lengths exhibits the strength of REEF in comparison to SPADE. For input lengths of 4,6, and 8, REEF manages to find a much higher percentage of words than SPADE. Clearly, for text REEF performs much better mining than SPADE and the sequences mined are more meaningful. Although the runtime for SPADE was shorter than for REEF the tradeoff between runtime and output quality is clearly illustrated on the textual data. For many data sets, as for TEXT, it is worth spending more time to the more meaningful sequences in the mining process.

VI. Conclusion and Future Work

We developed an algorithm for frequent sequence mining named REEF that overcomes the short sequence bias present in many mining algorithms. We did this by defining *norm-frequency* and using it to replace support based frequency used in algorithms such as SPADE. In order to ensure scalability of REEF we introduced a bound for pruning in the mining process. This makes the runtime for REEF comparable to that of SPADE.

The use of the bound requires a preprocessing stage to calculate statistics on a sample of the data set. As this sampling creates a distortion in the sampled measures, we present a method to correct this distortion.

Our extensive experimental evaluation is performed on four different types of data sets. They are a mixture of synthetic and various real world data sets, thus providing a broad performance analysis of REEF. Our experimental results show without doubt that the bias is indeed eliminated. REEF succeeds in finding frequent sequences of various lengths and is not limited to finding short sequences. We show the scalability of REEF and addressed the tradeoff between runtime to quality



(a) Lewis Carroll (b) Shakespeare (c) Linux Guide

Figure 11: Percentage of real words found among sequences.

of mined sequences. We illustrated that REEF produces a more variant distribution of output pattern lengths. We also clearly showed on textual data how REEF mines more real words than SPADE. This seems to indicate that when mining sequences are not textual, we can expect to mine meaningful sequences as well. Although REEF requires slightly longer runtime than SPADE the nature of the mined sequences makes this worthwhile. In the future, we hope to improve the bound used for mining. Thus, providing an algorithm that is more efficient while still producing the high quality sequences we found in REEF.

REFERENCES

[1] A. Richardson, G. A. Kaminka, and S. Kraus, "Reef: Resolving length bias in frequent sequence mining," in *IMMM 2013, The Third International Conference on Advances in Information Mining and Management*, 2013, pp. 91–96.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, 1995, pp. 3–14.

[3] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovering frequent episodes in sequences (extended abstract)," in *1st Conference on Knowledge Discovery and Data Mining*, 1995, pp. 210–215.

[4] F. Elloumi and M. Nason, "Searchpattool: a new method for mining the most specific frequent patterns for binding sites with application to prokaryotic dna sequences." *BMC Bioinformatics*, vol. 8, pp. 1–18, 2007.

[5] N. Zhong, Y. Li, and S.-T. Wu, "Effective pattern discovery for text mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 1, pp. 30–44, 2012.

[6] W. Fan, M. Miller, S. J. Stolfo, W. Lee, and P. K. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowledge and Information Systems*, vol. 6, no. 5, pp. 507–527, 2004.

[7] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proc. 20th NIST-NCSC National Information Systems Security Conference*, 1997, pp. 366–380.

[8] C.-H. Lee and V. S. Tseng, "PTCR-Miner: Progressive temporal class rule mining for multivariate temporal data classification," in *IEEE International Conference on Data Mining Workshops*, 2010, pp. 25–32.

[9] P. Senkul and S. Salin, "Improving pattern quality in web usage mining by using semantic information," *Knowledge and Information Systems*, no. 3, pp. 527–541, 2011.

[10] Y. Horman and G. A. Kaminka, "Removing biases in unsupervised learning of sequential patterns," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 457–480, 2007.

[11] C. Luo and S. M. Chung, "A scalable algorithm for mining maximal frequent sequences using sampling," in *ICTAI '04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 156–165.

[12] U. Yun, "An efficient mining of weighted frequent patterns with length decreasing support constraints," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 741–752, 2008.

[13] M. Seno and G. Karypis, "SLPMiner: An algorithm for finding frequent sequential patterns using length-decreasing support constraint," in *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2002, p. 418.

[14] P. Tzvetkov, X. Yan, and J. Han, "Tsp: Mining top-k closed sequential patterns," *Knowledge and Information Systems*, vol. 7, no. 4, pp. 438–457, 2005.

[15] C. Luo and S. Chung, "A scalable algorithm for mining maximal frequent sequences using a sample," *Knowledge and Information Systems*, vol. 15, no. 2, pp. 149–179, 2008.

[16] N. Tatti and B. Cule, "Mining closed strict episodes," in *ICDM '10: Proceedings of the 2010 Tenth IEEE International Conference on Data Mining*, no. 1, 2010, pp. 34–66.

[17] N. Tatti, "Maximum entropy based significance of itemsets," *Knowledge and Information Systems*, vol. 17, no. 1, pp. 57–77, 2008.

[18] G. I. Webb, "Self-sufficient itemsets: An approach to screening potentially interesting associations between items," *ACM Trans. Knowl. Discov. Data*, vol. 4, no. 1, pp. 1–20, jan 2010.

[19] M. J. Zaki, "SPADE: An efficient algorithm for mining frequent sequences," *Machine Learning Journal*, vol. 42, no. 1/2, pp. 31–60, 2001.

[20] A. Richardson, G. Kaminka, and S. Kraus, "CUBS: Multivariate sequence classification using bounded z-score with sampling," in *IEEE International Conference on Data Mining Workshops*, 2010, pp. 72–79.

[21] G. A. Kaminka, M. Fidanboylu, A. Chang, and M. M. Veloso, "Learning the sequential behavior of teams from observations," in *RoboCup 2002: Robot Soccer World Cup VI*, ser. LNAI, G. A. Kaminka, P. U. Lima, and R. Rojas, Eds. Berlin: Springer-Verlag, 2003, no. 2752, pp. 111–125.

[22] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *EDBT '96 Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, 1996, pp. 3–17.

[23] A. Salam and M. S. H. Khayal, "Mining top-k frequent patterns without minimum support threshold," *Knowledge and Information Systems*, 2011.

[24] C. H. Mooney and J. F. Roddick, "Sequential pattern mining – approaches and algorithms," *ACM Comput. Surv.*, vol. 45, no. 2, pp. 19:1–19:39, Mar. 2013.

[25] M. Seno and G. Karypis, "LPMiner: An algorithm for finding frequent itemsets using length-decreasing support constraint," in *ICDM '01 Proceedings of the 2001 IEEE International Conference on Data Mining*, 2001.

[26] M. Bilenko, S. Basu, and M. Sahami, "Adaptive product normalization: Using online learning for record linkage in comparison shopping," in *Proceedings of the Fifth IEEE International Conference on Data Mining*,

ser. ICDM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 58–65.

[27] M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara, "Evaluation of sampling for data mining of association rules," in *RIDE '97 Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications*, 1997, pp. 42–50.

[28] C. Raissi and P. Poncelet, "Sampling for sequential pattern mining: From static databases to data streams," *ICDM '07: Proceedings of the 2007 IEEE International Conference on Data Mining*, vol. 0, pp. 631–636, 2007.

[29] H. Toivonen, "Sampling large databases for association rules," in *Proceedings of the 22th International Conference on Very Large Data Bases*, ser. VLDB '96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, pp. 134–145.

[30] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0.

[31] R. Agrawal, M. Mehta, J. Shafer, and R. Srikant, "The QUEST data mining system," in *Proc. of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining*, 1996, pp. 244–249.

[32] L. Carroll, "Alice's Adventures in Wonderland," Project Gutenberg, [Accessed Jan. 15, 2009]. [Online]. Available: www.gutenberg.org/ebooks/11

[33] W. Shakespeare, "A Midsummer Night's Dream," Project Gutenberg, [Accessed Jan. 15, 2009]. [Online]. Available: //www.gutenberg.org/ebooks/2242

[34] J. Goerzen and O. Othman, "Debian gnu/linux : Guide to installation and usage," Project Gutenberg, [Accessed Jan. 15, 2009]. [Online]. Available: www.gutenberg.org/ebooks/6527

[35] A. E. Ahmed, "A new biometric technology based on mouse dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 3, pp. 165–179, 2007.

[36] R. Janakiraman and T. Sim, "Keystroke dynamics in a general setting," in *Advances in Biometrics*, ser. Lecture Notes in Computer Science, vol. 4642/2007. Springer Berlin/ Heidelberg, August 30 2007, pp. 584–593.

[37] C. Earl, S. Patrick, and P. I. A. Lloyd, "Fuzzy logic based viewer itentification," Patent No. WO/2007/131069, 2007.

[38] B. Sigurd, M. Eeg-Olofsson, and J. V. Weijer, "Word length, sentence length and frequency zipf revisited," *Studia Linguistica*, vol. 58, no. 1, pp. 37–52, 2004.

ALPACA: A Decentralized, Privacy-Centric and Context-Aware Framework for the Dissemination of Context Information

Florian Dorfmeister, Sebastian Feld, and Claudia Linnhoff-Popien
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität München
Munich, Germany
Email: {florian.dorfmeister, sebastian.feld, linnhoff}@ifi.lmu.de

Abstract—**With the ongoing rise of smartphones as everyday mobile devices and their steadily increasing amount of sensing and communication capabilities, we are on the brink of a subtle, widespread adoption of context-aware computing techniques into our daily lives. Focusing on functionality and performance, the majority of existing architectures for managing context information typically deploy central components for collecting, analyzing and distributing its users' up-to-date data. However, preservation of users' privacy needs remains a crucial factor for such systems' acceptableness. Inspired by existing works on privacy in context-aware applications and the authors' beliefs in the necessity to put users back in control, this article adopts a privacy-centric perspective and presents ALPACA: A novel approach for modeling and managing a user's rich context information in a user-centric and privacy-preserving way fit for a multitude of different usage scenarios. To this end, this article offers a general conceptual mapping of a user's privacy needs to distinct layers. Based on this conceptualization we introduce a privacy-centric approach for modeling this information. Additionally, we propose a context-aware mechanism for the definition of context-dependent release triggers in order to enable fine-grained control over the disclosure of sensitive information. Finally, we present the components of the proposed system architecture, explain how they interact with each other and discuss how our framework can be integrated into a modern mobile operating system.**

Keywords-context modeling, context-awareness, privacy-centric design, context-dependent privacy policies, context obfuscation.

I. INTRODUCTION

Both the acquisition of a user's current context, e.g., by applying activity classification or other reasoning techniques to her smartphone's sensor readings as well as approaches for effectively modeling and managing context information are active areas of research. At the same time, however, also the leakage of sensitive personal data is an issue of substantial interest on both a legal, social, and technical level. The ongoing scandalization of agencies' large-scale data collection on the Internet hopefully affords an appropriate, albeit unpleasant, opportunity to sensitize the public for these privacy issues. The latter get additionally tightened as, due to the popularity of smartphones, we are heading towards a full supply of small electronic devices with broadband Internet access, extensive sensing and computing capabilities. The privacy problems naturally exist regardless of extensive eavesdropping, as the number of applications that silently collect context information and send it to the Internet is legion. Privacy-aware users usually face an all-or-nothing option at install time and, once approved, there is no control over an application's usage of personal or context information at all. In order to contribute to the tackling of some of the technical aspects of these problems, we originally published the concepts and design of the *Layered Architecture for Privacy-Assertions in Context-Aware Applications* (ALPACA) in [1]. The main goal of this privacy-centric framework for managing context information is to turn a user's mobile device into a personal data vault that only the user has full access to as well as to enable fine-grained access control mechanisms for the dissemination of sensitive information. In this article, we will provide an updated and more thorough view at the different components and present new additions to the framework. In particular, we will go into detail on our ontology-based modeling approach and the context-aware trigger mechanism, which allows for the definition of context-dependent privacy policies. We will give more insight into our system's key components, communication protocol, and entity authentication and outline its applicability to the Android operating system and different usage scenarios.

One can think of many beneficial use cases for context-aware applications, such as proactive route planning services taking into consideration the current traffic volume and a user's appointment schedule, smart mechanisms automatically adjusting a phone's audio profile based on the user's current occupation or context-aware online social networking, e.g., buddy finder apps notifying the user about friends in proximity [2], [3]. In addition, there are applications such as the *SmartBEEs* context-aware business platform [4], which do not act based on a single-user or peer-to-peer basis, but leverage the combined knowledge of multiple users' current contexts and their surroundings' state, e.g., for business process optimization. In order to prevent unintentional disclosure of personal data, users must yet be able to control what kind and resolution of context information applications are able to collect at any time. Enabling the acquisition of a user's context information with the help of her smartphone's sensors and eligible reasoning mechanisms in return also enables spying on this person. Thereby "one person's sensor is another person's spy", as [5] puts it. For the enabling of high levels of service quality, the algorithms used for context acquisition typically aim at maximizing resolution, freshness, and accuracy of their findings. When talking about preserving a user's privacy, however, different and partly even contradicting objectives are to be pursued. For example, in many situations it might be perfectly sound to deliberately reduce the resolution of a piece of context information before sharing it with others in order not to reveal too much. Many approaches for managing context information focus on the generation, modeling, processing, and efficient distribution of the latter as well as the realization of context-aware applications built thereon. Most of these works take a primarily functional view on their system and try to maximize parameters such as classification accuracy, availability, and scalability. Some works also argue that not only the usability and utility of context-aware applications are paramount for a wide acceptance, but also the establishment of appropriate privacy mechanisms, which make users feel safe and comfortable within such ubiquitous computing environments. Concordantly, in this work we propose an integrated approach for giving the full control over the release and granularity of sensitive context information to the data's owner, comprising a novel ontology-based and privacy-centric context model as well as a mechanism for defining context-dependent access control policies and the corresponding system architecture.

The remainder of this article is structured as follows: In Section II, we will give our problem statement and a definition of sensitive context. Section III reviews related work on privacy in context-aware applications and presents a comprehensive list of requirements for privacy in context-aware applications. In Section IV, we present a general conceptualization of a user's privacy needs, which is the foundation of our privacy-centric modeling approach presented in Section V. In Section VI, our context-aware release triggers for the definition of a user's privacy policies will be introduced. Our framework's system architecture and communication protocol are described in Section VII. After discussing our approach in Section VIII, we conclude.

II. Problem Statement

This work focuses on the modeling and management of a mobile user's rich context information in a privacy-preserving way. We found our understanding of context on the general definition by Abowd et al., declaring context to be "*any information, that can be used to characterize the situation of an entity* [...] *relevant to the interaction between a user and an application*" [6]. Strictly following a user-centric approach, we assume a user's smartphone to be the primary source of information about her current situation. With this work focusing not only on functionality, but on protecting a mobile device user's privacy, we complement the given definition to characterize *sensitive context* as follows: *Sensitive context is any information available through a user's device, that can be used by any entity to infer the situation of the user regardless of any interaction between the user and any application*. Given this definition, we aim at designing a generic and integrated solution for the management and context-dependent access control of a mobile device user's static and dynamic context information. Consequently, we consider it essential to primarily focus on putting the user in full control over the acquisition, release, and resolution of her personal data.

From a privacy-centric point of view, on the one hand, the less data is going to leave the user's mobile device, the better. Yet in order to enable a diversity of context-aware applications, there is usually a need for communicating one's current context information to other parties. For privacy reasons, however, we argue that there must not be any party but the user herself able to access or control her complete context information at any point in time, thereby ruling out any solutions based on a trusted third party approach. On the other hand, providing central reasoning components with some carefully selected context information seems nonetheless desirable in some situations, e.g., in order to allow for the efficient realization of multi-subject context-aware applications. Existing frameworks typically comprise four consecutive stages to enable context-awareness, i.e., context acquisition, context modeling, context exchange, and reasoning [7]. In order to protect a user's context privacy along this chain, an integrated approach should hence comprise the following requirements:

1) An *abstract conceptualization* of a user's privacy needs, that models context on easily comprehensible layers according to different groups of requesters.
2) A *formal model* of a user's context, that already integrates some core aspects of privacy of context information and provides the ability to easily integrate current and future privacy and security methods into context-aware applications.
3) An effective mechanism for controlling the dissemination in form of *user-defined privacy policies*. These policies have to consider the user's as well as the requesting entity's context and should be evaluated in a context-aware manner.
4) An *integrated system architecture*, that is suitable for the above mentioned use cases and all kinds of context requesters, i.e., applications running exclusively on the user's device, peer-to-peer, and third party services while always emitting only the minimum amount of information required.

In the next sections, we propose a solution to all of these requirements: The privacy layers described in Section IV incorporate an abstracted view on a user's privacy needs. Our *Context Representations* model is concerned with the formal representation of a user's context information (Section V). We design a mechanism for context-dependent privacy preferences in Section VI and eventually put the pieces together with our privacy-centric and context-aware framework for the dissemination of context information in Section VII.

III. Related Work

This section presents related work on privacy mechanisms for context-aware applications. Several different categories of approaches can be found in literature. Some of them rely on trusted third party (TTP) solutions for efficient context dissemination, whereas other systems adopt a peer-to-peer (P2P) based approach in order to avoid such central points of attack. In addition, there are rule languages for defining access control based on contextual information as well as different obfuscation techniques for adequately reducing the richness of a user's context information before their release.

A. Context management frameworks and tools

A common architectural model for the realization of location-based and context-aware applications is the use of a TTP acting as some kind of middleware for the aggregation of its users' context information [8]. It is necessary for all of the system's participants to fully trust this component. The CoPS architecture introduced in [9] implements such a central privacy service. While allowing for different granularities of context items, it does not permit the definition of context-dependent access rules. Another TTP-based approach called CPE [10] enables the definition of context-dependent privacy preferences, but lacks mechanisms for releasing information in different granularities. With a focus on context-dependent security policies, the CoBrA platform [11] deploys the Rei policy language [12] in order to enable the definition of access control rules depending on a user's current context. Beyond that there is much literature on different techniques for the obfuscation of contextual information. As an example, [13] presents another centralized approach focussing on context ownership and offering obfuscation mechanisms for several kinds of context information based on SensorML process chains, obfuscation ontologies, and detailed taxonomies describing dynamic granularity levels. In contrast to these systems, purely P2P-based approaches such as [14] get along without any central component. As a major drawback, such architectures can hardly be efficiently deployed in applications depending on up-to-date context information of a whole group of users at the same time. Behrooz et al. presents CPPL, which is a policy language that can be used to define the access rules for a user's context information in a context-dependent way [15]. Noticeable, the rules that can be set up here are not only context-dependent, but the engine responsible for policy enforcement itself is context-aware, i.e., it maintains an up-to-date set of active policies according to the user's current context in order to minimize the number of policies that have to be considered when processing an incoming request for context information. However, the policy mechanism has been designed to only depend on the context owner's content and its social relationship to the context requester. We argue, however, that a comprehensive policy approach should be able to also take the requester's current context into account as well. Consequently, it is not clear how a requester's context could be transferred to the context owner in a sensible and privacy preserving way, too.

B. Privacy mechanisms for context-aware applications

Based on the above mentioned works [9]–[15] and further literature on privacy in context-aware applications, in this section we identify and complement a set of different techniques and requirements for realizing different aspects of privacy. Naturally, an integrated approach for modeling and managing context should incorporate all of these mechanisms. In the following, we will list and briefly explain the most important of these requirements.

1) Fine-grained control: Above all, a minimum set of **access control** operators such as *grant* and *deny* has to be available in order to be able to define different requesters' access rights. Additionally, users should be in a position to define their privacy preferences in a **context-dependent** way. For example, Xie et al. [16] show that a user's willingness to release any of her context information to others depends on a number of different factors, such as time, companion and emotion. Considering the nuances in privacy preferences among different users, Bokhove et al. also argue for the need of **fine-grained** and **expressive** privacy mechanisms [17].

2) Adjustable quality-of-context (QoC): **Variable granularity** and other QoC-related aspects can be used to reduce a piece of context information's precision or accuracy [18]. As an example, consider a user reading her e-mails. Different granularities of her currently modeled activity context might contain `read-e-mail`, `computer-work`, `office-work` and `working`. Moreover, credibility issues, such as **intentional ambiguity** and **(white) lies**, can be used in order to lower the confidence or validity of a piece of context information. Notice that such behavior is common in our everyday actions in the offline world, too, e.g., not answering telephone calls in order to conceal presence or availability. **Adjustable freshness** and **temporal resolution** are other means for intentionally reducing a piece of information's quality, e.g., regarding its age, capturing time or temporal validity. It can hence be used in a similar way as intentional ambiguity to obfuscate a user's context.

3) Consistency and completeness: We define **consistency** of a user's privacy preferences as another important requirement, stating that a context requester must not be able to retrieve ambiguous, e.g., contradicting pieces of context information. Considering that completely denying a request for a piece of a user's context information might itself reveal much, we define **completeness** to be the principle of answering any request with a plausible response. This is important, as it is often more privacy-preserving to give an imprecise yet plausible reply than to refuse a request.

4) Notifications and logging: **Notifications** can optionally be sent to a user upon a request of her context information [9]. Additionally, requests should be **logged**. These techniques allow for users being well informed about usage of their personal information [17] and can be used as a social means able to contain the intentional abuse of contextual information.

5) Symmetry: Another important concept adapted from the offline world is **symmetry**, stating that a certain party has to reveal just as much of its own information as it requests [19]. In connection with context-aware applications, this especially applies to the realization of peer-to-peer-based scenarios.

6) Anonymity and pseudonymity: Other concepts for privacy in context-aware applications are **anonymity**, **pseudonymity**, and **k-anonymity** [20] relating to a user's identity not being known by a system, only being known by a pseudonym, and a user being indistinguishable from k-1 other users, respectively.

In order to seamlessly protect a user's privacy, it also seems beneficial to closely band together the acquisition, modeling and management of a user's context information. More general requirements are extensibility, e.g., a system's ability to automatically integrate new types of context sources and usability.

IV. A Layered Conceptualization of Privacy

We will now introduce a practical conceptualization of privacy as a general mapping from different groups of requesters of context information to distinct privacy layers. These layers resemble what we believe to be a good compromise about a privacy-aware user's sensation of different levels of information accessibility and reduction of complexity. To keep

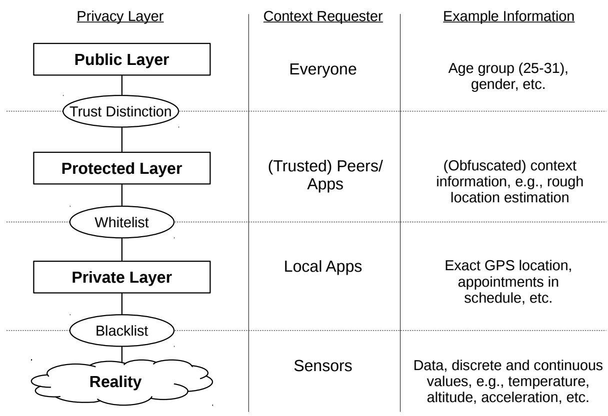


Figure 1. The four different privacy layers defined in ALPACA and their most likely audiences as well as some example context items for each layer.

things small and simple, there are only two roles defined: the *context owner*, who is the user whose context is to be protected and the *context requester*, which might be any application, service or peer requesting access to the user's context.

As shown in Figure 1, we define four logical layers that can be mapped to a user's privacy needs, as well as different gateway mechanisms controlling these layers' permeability. At the bottom layer we put *reality*, possibly containing more information than any types of sensors and reasoning mechanisms will ever be able to capture. Obviously, there is no need to implement anything on this layer. However, it still has to be considered as this layer constitutes what is to be reflected by any context modeling approach. Users might feel uncomfortable knowing that each of their smartphone's sensors is recording data all the time, e.g., in some situations or at certain locations a user might not want her smartphone's microphone to record audio, and hence this sensor should be turned off automatically. Users should thus be able to set up a context-dependent blacklist for defining which sources of context information should be turned off. Every time the user's context changes, this blacklist has to be re-evaluated.

The next layer is the *private layer*, holding all the information a user wants to have available for herself, i.e., context-aware services and applications running exclusively on her mobile device. Consider for example locally run apps, which adapt their appearance and behaviour according to the user's current context. In order for such services to be responsive and proactive, this layer enables access to the most fresh and sophisticated context representations. Naturally, these high-resolution representations are probably not intended for everyone else as well. Thus, the trigger mechanism described in Section VI is used as a context-dependent whitelist for the release of certain representations to the upper layers.

Context information which pass this whitelist enter the *protected layer* and might hence be available for some other entities, too, such as trusted services and peers. For example, a user might be reluctant to share her current whereabouts with everyone, but maybe with some of her friends in her spare time or with her employee during working hours. Naturally, the number and composition of context representations available on this layer will hence change dynamically based on the user's privacy preferences and current context.

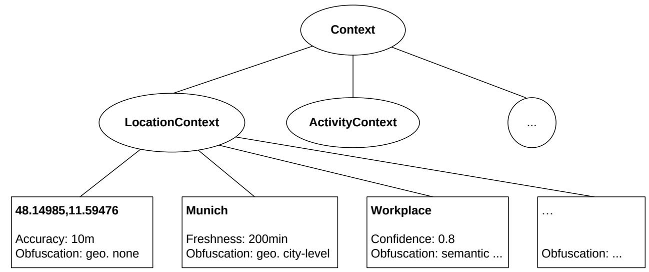


Figure 2. An example context tree with its second-level *Context* nodes linking to an arbitrary number of representations, each indicating its level of obfuscation based on the semantics of the underlying information.

Additionally, a user might be willing to share some kind of information about herself with anyone, meaning that these information are available on the *public layer*. This might, e.g., be true for information that are somehow obvious anyway, such as personal profile data containing the user's gender or age group. However, a user might still define notifications to be displayed when these kinds of information are being requested. That said, notice that our system's user is of course not forced to abide by these layers in the way we just described, but rather can individually choose which level of visibility fits her own situation by the use of appropriate release triggers.

V. Modeling Context Using ALPACA-CoRe

A privacy-oriented approach for modeling a user's context should inherently focus on the enabling of fine-grained access control mechanisms over these dynamic and sensitive data. Applying the privacy techniques listed in Section III to the modeling part of our framework implies that at least the concepts of variable granularity, intentional ambiguity, white lies, freshness, and consistency should directly be integrated into our model. Our basic concept is that a user should be able to maintain different versions of her current context information that can be shared with different context requesters, as the latter might not appear equally trustworthy to the context owner. Accordingly, *Context Representations* (CoRe) model is designed to inherently store several heterogenous versions of a user's context information in parallel, each serving a different purpose for different groups of context requesters. Notice, that there have been some considerable modifications to the original version presented in [1].

A. Representations of context information

Approaches for context modeling (cf. [21] for a comprehensive survey) typically adopt a hierarchic understanding of context, which can be illustrated using a tree-based structure. In such a hierarchic scheme, the root node aggregates all categories of a user's context information. At the second level, distinctions between the basic types of context information are made, such as a user's current time, location, or activity. Each of the tree's second-level nodes may be a parent to a hierarchy of an arbitrary number of nodes representing the corresponding category in a different way. To preserve a user's privacy, the simultaneously possible representations of a user's current location might differ, e.g., in their spatial resolution, accuracy and freshness, or in case of a white lie maybe even validity. Apart from the creation of such hierarchies being a

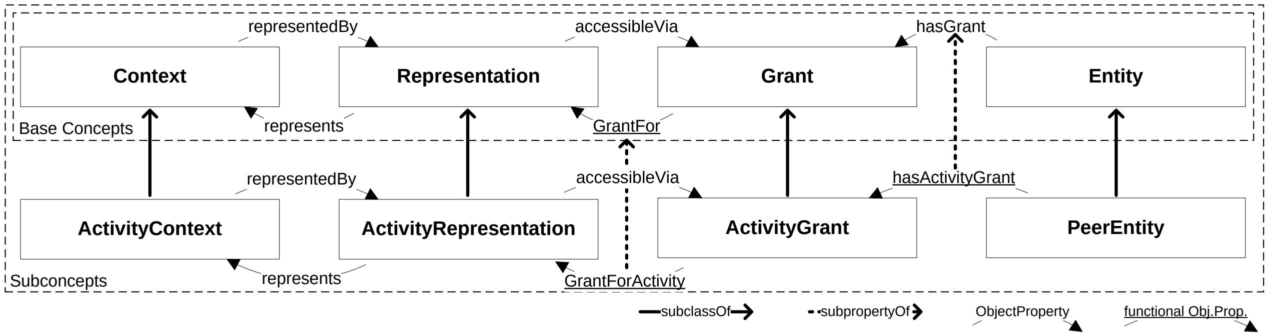


Figure 3. The cornerstone concepts and properties of the CoRe model: *Context*, *Representation*, *Grant*, and *Entity* as well as corresponding example subclasses.

non-trivial task [13], it is thus not possible to stick to such taxonomies in general, due to contemporaneous representations of the same kind of context differing in more than one dimension. In contrast to existing context models such as MUSIC [22], which use the term "representation" in order to label the data formats used for communication (such as XML or JSON), we hence suppose distinct representations of context information to possibly differ from each other on a semantic level, independent from any encoding. As an example, consider the three different representations of the user's current location in Figure 2: The instance on the left holds the current GPS position fix of the user. In contrast, the one in the middle only states the user's location on a city level, while the third uses a non-geographic, symbolic location identifier that cannot be mapped to a geographic one – at least not without any further knowledge about the user. The acquisition of these representations is not within the scope of this work. However, we require this process to be performed by sensing, reasoning, and context obfuscation services running on the user's device. The different representations stored in the model can be marked to be accessible by different groups of requesters. Which representation is to be released to whom might depend on the privacy level assigned to the requester and on context information itself: Services running exclusively on the user's mobile device are likely to be allowed to access the user's context information with the highest resolution available, whereas peers and third party services might only be able to retrieve some kind of adequately obfuscated representations.

B. Components of the CoRe model

Following an OWL-based modeling approach, our context model consists of the base classes shown in Figure 3, i.e., *Context*, *Representation*, *Grant* and *Entity*. Each of the first three has subclasses for the different categories of context, such as *ActivityContext*, *ActivityRepresentation* and *ActivityGrant*, respectively. As stated above and indicated by the *representedBy* property, a certain subclass of context may be described by multiple contemporaneous instances of the according subclass of *Representation*. The model's basic structure was inspired by the ASC model by Strang et al. [23] enabling service interoperability based on a shared understanding of and transformation rules for different, yet logically equivalent scales. There, the interrelation of different scales such as a mile and kilometer scale are described along with transformation rules that can be used to translate from one scheme to another. Quite the contrary, our CoRe approach aims at modeling different representations of the same kind of context information, which according to the privacy mechanisms listed in Section III do not necessarily have to share a similar or at least consistent meaning at all. In fact, there must not exist any transformation rules which allow for a trivial conversion, e.g., from a low-resolution representation to a high-resolution one.

As an enabler for the definition of privacy preferences based on our model, an instance of *Representation* can be made accessible to a group of context requesters via subclass instances of *Grant*. This can be modeled by using the corresponding subproperty of the functional *GrantFor* property as depicted in Figure 3. A context requester can be any *Entity* requesting some of the user's context information and can be referred to individually, by group membership, or by dynamic constraints on context. In order to allow for privacy policies that also depend on the requester's context, such as "friends at my location", the CoRe model also stores the known contexts of all active requesters. As claimed by the requirement for consistency, however, a requesting entity must never be granted access to more than one representation of the same type of context information at the same time, as this might result in an ambiguous response. In order to integrate this understanding of consistency into the CoRe model all subproperties of the *hasGrant* property are marked as *functional*, too. Constricting the modeled relations by the ontology's *functional property* constraint as described allows for dynamic consistency checking at runtime based on the built-in OWL reasoning capabilities. Hence, a user's privacy policies do not have to be tested statically for any inconsistencies, which instead can automatically be detected by an inconsistent state of the ontology. Likewise, we provide the *GrantFor* property with a *minCardinality* constraint of 1, stating that there must also be at least one *Representation* for each *Grant*. This can be used to realize some kind of garbage collection that is able to automatically detect and remove an obsolete *Grant*. Notice, that the inverses of these properties are not constrained like that, e.g., one instance of *Representation* can be accessible via several *Grant* instances simultaneously and a *Representation* can exist even if there is no corresponding *Grant*.

In order to be able to automatically assess the resolution, validity, and sensitivity of a *Representation* instance, each of them provides information about its *ObfuscationLevel*. As shown in Figure 4, each type of context can have an arbitrary number of obfuscation levels and each *Representation* belongs to exactly one of these. Considering the great differences in semantics that different types of context information and obfuscation techniques might display, a generic labeling ap-

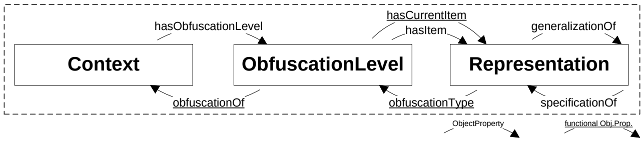


Figure 4. Each instance of *Representation* describes a certain type of *Context* using a corresponding scheme, i.e., a certain instance of *ObfuscationLevel*.

proach for these levels seems unfeasible. Hence, each subclass of *Context* is expected to define its own obfuscation scales, probably provided by the service used for creating the corresponding representations. Within a single scale there might be a naturally defined hierarchy, such as the granularity of location. Instances of the same subclass of *Representation* can hence be marked to be a *generalizationOf* one another, which is a *transitive* property. This can be used to model that a context requester who is granted access to a high-resolution representation can also access low-resolution ones, e.g., if she only asks for the latter. This does not violate the consistency requirement, as conflicting responses caused by contradicting representations cannot arise. In addition, each *ObfuscationLevel* has a functional *hasCurrentItem* property indicating which of the corresponding *Representation* instances is currently valid and should thus be used for the evaluation of the user's privacy policies. This is necessary as for each *ObfuscationLevel* there might be a multitude of *Representation* instances stored in the model in parallel, some of which are not valid any more, but still remain in the CoRe model, e.g., to enable adjusting the freshness of released information.

In this section, we have described a novel, privacy-centric approach for modeling a user's context information, which can be used to manage multiple representations of a user's context information intended for different context requesters. In the next section, we will focus on how context-dependent privacy policies can be realized based on our model.

VI. Context-Dependent Privacy Policies

In order to protect a user's context information from unintended leakage, our system takes a highly restrictive, whitelist-based approach: Apart from fully trusted apps running exclusively on the user's device, explicitly stating that some information should be accessible by a certain requester is the only possibility for a user to share any piece of context information. In this section, we will introduce a suitable mechanism for the realization of corresponding privacy policies, show example usage, and demonstrate our framework's capability to preserve consistency by automatically detecting and resolving conflicting policy definitions at runtime.

A. Trigger-based release of context information

The ALPACA framework comprises a new mechanism for the definition of flexible, fine-grained, and context-based access control rules on behalf of the context owner. In order to facilitate the definition of both static and context-dependent privacy policies, the context owner must be able to manage her contacts in groups comparable to current online social networking (OSN) services. To this end, *Release Triggers* (RTs) can be set up as privacy policies for controlling the

```
<releaseTriggerId>

Conditions*
cond(CO, context_type, representedBy,
obfuscation_level, …)

Effects+
grant({(CR, context_type, representedBy,
obfuscation_level, …)[AND (CR,…)]}
-> {(CO, context_type, representedBy,
obfuscation_level, …)[AND (CO,…)]})

Metadata*
precedesOver*, sideEffect*
```

Figure 5. The logical structure of the release trigger mechanism: *Conditions* and *Effects* allow for the definition of context-dependent privacy preferences.

release of a user's context information. Behrooz et al. differentiate between situational context, that can be used to evaluate privacy policies and sensitive context, that is to be protected by the policies [15]. In contrast, we argue that all of a user's context information is likely to be both relevant in the policy definitions and worthy of protection. The RT concept hence enables the deployment of *context-dependent* privacy preferences based on and intended for the rich context information stored in the CoRe model. It can be used for dynamically whitelisting the release of a certain *Representation* of context information to a context requester.

The logical structure of a trigger and its subcomponents are outlined in Figure 5. For identification purposes, every trigger is assigned a unique name. It further consists of sets of *Conditions* and *Effects* as well as a *Metadata* section. The *Condition* part holds all prerequisites posed on the context owner's (CO) current context for the trigger to become active. Here, several conditions can be specified per trigger, which are evaluated as a whole using the logical `AND` operator. The required conditions may base on *Representations* of any context type and *ObfuscationLevel* that are available in CoRe. This allows the definition of conditions based on any desired granularity as well as any combinations of context types. The *Effects* part states what kind of *Representation* is to be released upon a certain entity (CR) requesting context information. In order to differentiate between context requesters in the *Effects* part, entities may either be referred to explicitly or implicitly via group membership, context properties, or any combination thereof. The latter allows for referring to the requester's context in a trigger definition, too, so that the final release of context information can depend on the requesting entity's context. In the *Metadata* part, e.g., information about the relative importance of a trigger can be defined, such as precedence rules used for conflict resolution.

If a trigger fires, the corresponding instance of *Representation* will be made accessible to a context requester. In terms of the concepts of the CoRe model, this is achieved by the creation of a new *Grant* instance. The latter will be linked to the *Entity* characterized in the release trigger's *Effects* part as well as with the described *Representation*. Which instance of *Representation* is to be released to the requester can be defined by the context owner in terms of the desired obfuscation level, maximum or minimum accuracy, confidence, and freshness properties. Again, several effects can be defined within the scope of a single trigger. However, it is important that each release of a *Representation* by any *Entity* is managed by a dedicated *Grant* instance, as the access to individual representations may depend on different conditions on the requester's context. From [9] we have adapted the idea that accessing a piece of context information by a certain requester might have side effects such as notifying the user, as we agree on that being a proper means suitable for keeping the user informed and containing data abuse. If desired, a user can hence also specify which *SideEffect* should be activated when a certain kind of *Representation* is accessed using a certain *Grant*. The metadata section can be used to define the relative importance of a trigger compared to others, which is needed for resolving conflicting policy definitions. Both the side effects and precedence rules will also be added to the CoRe model.

B. Context-aware policy evaluation

Apart from allowing for context-dependent privacy policies, our approach can be considered *context-aware*, too, as the policy evaluation takes into account the user's current situation in order to reduce the number of rules that have to be evaluated upon an incoming request for context information. To this end, the evaluation of a user's release triggers is performed in two stages, thereby fulfilling the requirement for context-awareness of privacy policies stated in [15]. In the first stage, a release trigger is set active if all of its *Conditions* are met. As already explained, these conditions only depend on the context owner's locally known context and can hence be evaluated pro-actively to any context request. The up-to-date subset of active triggers is managed directly in the CoRe model and is adjusted accordingly with each change of the user's context. As a result, only active triggers have to be evaluated upon an incoming request for context information, i.e., during the second stage. This procedure can be naively implemented as follows: If the local context changes, all *Grant* instances are removed from the model. In the next step, all triggers are removed from the context owner's active trigger subset. Notice, that in this state no one but *private layer* entities has access to the context information. Now the first, context-aware stage of the trigger evaluation starts: Based on the currently modeled context, all triggers' *Conditions* will be matched in order to decide which triggers enter the active set. This set then holds all triggers that can potentially fire upon an incoming request for the user's context information. The second stage is a lazy rule evaluation performed in reaction to an actual request. Consequently, this evaluation only has to be performed on the filtered subset of active triggers instead of the user's complete policy set, which has the potential to improve the response time of the system. The active triggers will now be evaluated and in the case a trigger fires, a corresponding *Grant* instance as well as any *SideEffects* are added to the model.

To enable context-awareness in an actual implementation of the trigger mechanism in a rule language, the triggers' logical structure must be split into two parts: The *Conditions* part on the one hand and the *Effects* and *Metadata* part on the other are thus to be defined in two seperate rules, as illustrated by the example trigger definition in Figures 6, 7, and 8.

C. Example of a release trigger

We will now demonstrate the interplay of the CoRe model and our release trigger mechanism by means of a simple example scenario: Alice, who is the context owner, has set up a pair of release triggers defining which information she is willing to share with her colleagues. Her co-worker Bob is the context requester, who currently is at work. The full description of the first trigger can be seen in Figure 6 as pseudo-code. This

```
Trigger1 {
  Condition:
    Entity:           Alice
    Context:          TimeContext
    representedBy:    worktime
  Condition:
    Entity:           Alice
    Context:          LocationContext
    representedBy:    atWork
  Effect:
    Context:          LocationContext
    Obfuscation:      building-room
    ----
    Entity:           coworker
    Context:          LocationContext
    representedBy:    atWork
  Metadata:
    precedesOver:     Trigger2
}
```

Figure 6. Schematic definition of an example release trigger, *Trigger1*, depending on the context owner's current time and location given in pseudo code.

trigger defines that given it is worktime and Alice's current location is found to be her workplace, co-workers who are at work themselves are allowed to see in which room she currently resides in. The actual implementation of the trigger happens, as already mentioned, by outsourcing the condition part into an own rule. A second rule describes the effects and metadata part of the trigger. The corresponding sub-rules are shown in Figures 7 and 8, respectively. The syntax used here takes a generic format assuming rule evaluation in forward-chaining mode. The `makeTemp` creates a blank node in the model. For the sake of brevity, the definition of any SideEffects and usage of exact RDF syntax have been left out here.

The conclusion part of the *Conditions* rule

```
[shareRoomWithColleagues:
  (Alice hasContext ?t)
  (?t type TimeContext)
  (?x obfuscationOf ?t)
  (?x hasCurrentItem worktime)
  (Alice hasContext ?l)
  (?l type LocationContext)
  (?y obfuscationOf ?l)
  (?y hasCurrentItem atWork)
  -> (Alice hasActiveTrigger Trigger1)]
```

Figure 7. The context-aware *Conditions* part of *Trigger1* as a separate rule.

shareRoomWithColleagues shown in Figure 7 simply states that Trigger1 is to be added to Alice's set of active triggers. For this to happen, all conditions defined in the rule's body part must match the current state of the model. Just as required, the conditions of this rule exclusively depend on Alice's context information. So even though the overall result of the release trigger depends on the context of a requesting co-worker, too, shareRoomWithColleagues can be evaluated pro-actively to any incoming request. As soon as it is working hours and Alice arrives at work, Trigger1 will hence be set active. In contrast, as depicted

```
[Trigger1:
  (Alice hasContext ?l)
  (building-room obfuscationOf ?l)
  (building-room hasCurrentItem ?lr)
  (?lr type LocationRepresentation)
  (?cw isA coworker)
  (?cw hasContext ?cwl)
  (?cwl type LocationContext)
  (?cwl representedBy atWork)
  -> makeTemp(?g)
     (?g createdBy Trigger1)
     (?g GrantForLocation ?lr)
     (?cw hasLocationGrant ?g)]
```

Figure 8. *Effects* and *Metadata* parts of *Trigger1* as a separate rule.

in Figure 8, the Trigger1 rule itself takes into account the current context of the context requester. However, it only has to be evaluated when a context requester sent a request for her context information with his own context piggybacked. The first four triplets define that this rule targets the currently valid representation of Alice's location context on the *building-room* obfuscation level. The remaining lines of the rule's body state that the matching representation shall be made accessible for any co-worker who is at work: A new instance of *LocationGrant* is added to the model and linked to the chosen representation and matching entities. If there were any *SideEffects* that should be performed upon this *Grant* being used, it would also be stated here and the corresponding individuals are added to the model.

The release trigger shown in Figure 9 only relates to Alice's time context and does not pose any requirements to her whereabouts. In contrast to the first one, hence, this rule also matches when she is on a business trip or on a day off or simply late for work. Consequently, Alice decided to only let her colleagues know about her location on a city-level in this case. Assume that Bob requests her current location during the working hours: In case Bob is not at work himself, none of the two triggers will release her location information to him. If Bob is at work and Alice is not, only the second trigger matches the current situation and a *Grant* instance to her city-level location will be linked to Bob. When she is at work, however, both triggers match the situation. Accordingly, both rules fire and Bob is granted access to two different instances of *LocationRepresentation*, which clearly violates the requirement for consistency. How the CoRe model is capable of automatically detecting and resolving such conflicting policy definitions will be explained in the next section.

```
Trigger2 {
  Condition:
    Entity:          Alice
    Context:         TimeContext
    representedBy:   worktime
  Effect:
    Context:         LocationContext
    Obfuscation:     city
    ———
    Entity:          coworker
    Context:         LocationContext
    representedBy:   atWork
}
```

Figure 9. Schematic definition of an example release trigger, *Trigger2*, depending on the context owner's current time only.

D. Dealing with inconsistent privacy preferences

When making usage of context-dependent access control policies, it might happen that two or more (possibly contradicting) rules match a given situation. We define a set of privacy policies to be conflicting, if they produce an ambiguous set of access grants for a context requester. A set of *Grant* instances is ambiguous, if any *Entity* is granted access to more than one *Representation* instance of the same subclass of *Context* at the same time. This situation is prone to harm a user's integrity, which becomes evident when considering two contradicting representations, such as a user's true and fake location information: If a requester is allowed to access several representations of the same subtype of *Context* at once, the result is not only ambiguous, but also likely to negatively influence the context owner's respectability due to being caught lying. Considering the structure of the release trigger mechanism, such situations might occur when a context requester matches the entity description in more than one active trigger's *Effects*. In order to automatically detect such situations at runtime, we use the underlying ontology's built-in reasoning capabilites to dynamically check the CoRe model's state for consistency. Constraining all subproperties of the *hasGrant* and *GrantFor* properties to be *functional* and setting the *minCardinality* constraint of the *GrantFor* property to 1 as described in Section V-B allows for the following assertions:

1) Any instance of *Entity* is allowed to be linked to at most one instance of a given subclass of *Grant*.
2) Any *Grant* can only be linked to exactly one instance of the corresponding subclass of *Representation*.

If the rule evaluation produces a state that conflicts with these requirements, checking the model for consistency will result in an error. This check is performed each time the rule evaluation finishes. In order to solve conflicting policy definitions, the *precedesOver* property can be set for a release trigger. In an actual implementation, the first time an inconsistent state is detected the context owner can be notified about the conflict and the triggers that caused it. The user then can decide which of the triggers involved should precede over the other. The decision will be stored in a *precedesOver* property of the winning trigger. The next time the same inconsistent state arises, inferior *Grant* instances can be automatically removed from the model, as every *Grant* instance links to the trigger it has been created by.

In this section, we introduced the concept of *Release Triggers* for the context-based definition of a user's privacy preferences. The trigger mechanism has been designed to consider both the context owner's and optionally also the context requester's context to decide which kind of representation is to be released to the requester. The evaluation of these access control rules is performed in a context-aware way, which takes into account a user's context in order to minimize the number of rules that have to be evaluated upon incoming requests for context information. Eventually, we explained how conflicting policy definitions can dynamically be detected and resolved. In the next section, we will outline our framework's core components and overall system architecture.

VII. ALPACA System Architecture

In the previous sections, we introduced the privacy-centric CoRe approach for modeling a user's rich context information. Additionally, the RT mechanism enabled a context-dependent definition of a user's privacy policies. These aspects will now be joined together. For this purpose the ALPACA system is presented in the following together with a description of the communication occuring between the different components and context requesters. Finally, we sketch how our framework can be integrated into a modern mobile operating system such as Android.

A. Managing acquisition and release of context information

Our framework's key component is the *Privacy Manager* (PM), which is responsible for managing all access to the user's context information by enforcing the privacy policies set up by the context owner. To this end, the user is both able to control what information enters the CoRe model by blacklisting the acquisition of certain types of context information as well as to decide which kind of information is released to whom by whitelisting access to context information for trusted requesters (cf. Section IV). The overall system architecture of ALPACA and the logical placement of the PM is depicted in Figure 10. Serving as a gatekeeper, the PM is the only component able to directly access and update the information stored in the context model.

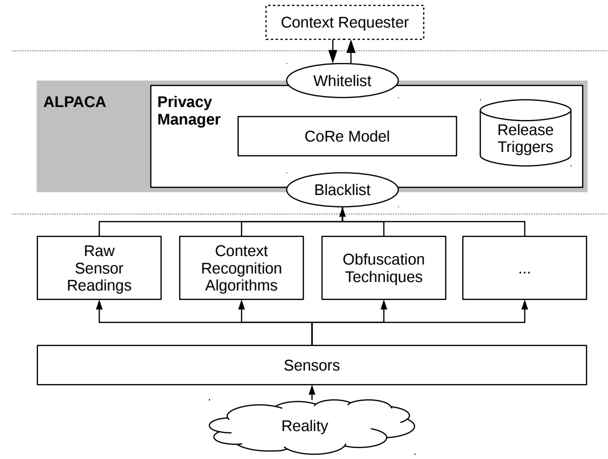


Figure 10. The *Privacy Manager* is the core component of the ALPACA framework and acts as an exclusive interface to the modeled context information for both context sources and context requesters.

For the acquisition of a user's up-to-date context information, we assume a number of different hardware and software sensors for low-level context recognition as well as services capable of high-level context reasoning to be available on a mobile device. In addition, we expect obfuscation techniques that can be used for adjusting the granularity or validity of a certain type of context information to be implemented as well. The corresponding sensors and services are responsible for capturing a user's context, transforming it into *Representation* instances of the corresponding subclass of *Context* and for providing the required meta data, such as *Freshness*, *Accuracy*, and *ObfuscationLevel*. Each time one of these context sources produces a new representation, it is reported to the PM. After authenticating the source, the PM will update the CoRe model by adding the *Representation* and setting the *hasCurrentItem* property of the respective obfuscation scale to link to the new *Representation*. As these operations alter the state of the context model, the PM now re-evaluates the context-dependent acquisition blacklist, turns corresponding sensors and reasoning services on or off, respectively, and re-evaluates the *Conditions* of the context owner's release triggers' in order to update the active set.

Moreover, the PM also controls the release of a user's context information to a context requester. As described in Section VI, this is realized on a per-request basis using lazy rule evaluation due to accessibility and granularity of a user's context information possibly depending on the requester's current context, too. A context requester can utilize most of the representations' meta data stored in the CoRe model to specify the characteristics of the requested *Representation*, such as context type, freshness, and accuracy. The PM will consider these requirements and look up the best matching accessible items. In the following, the basic communication flow within context requests issued by different context requesters will be explained.

B. Communication flow and request handling

The ALPACA framework aims at providing a generic solution for the management of a user's privacy preferences and the dissemination of context information fit for all kinds of usage scenarios. One can identify three different categories of context requesters, that can be mapped to the privacy layers introduced in Section IV as follows:

- *Local applications* running exclusively on the user's device and not sending any data off the device can be allowed to access a user's *private layer* context.
- *Third-party services* communicating context information to remote entities are to be placed on the *protected layer* or *public layer*. Distinct services might yet be granted access to different representations depending on the user's trust in the service provider.
- *Peer users* requesting the user's context information can be placed on the *protected layer*, too. Which representation they are allowed to see might depend on both the requesting peer's identity and context.

In order to enforce the privacy policies set up by the context owner, in each case the PM first verifies which entity is requesting context information and which of the above categories

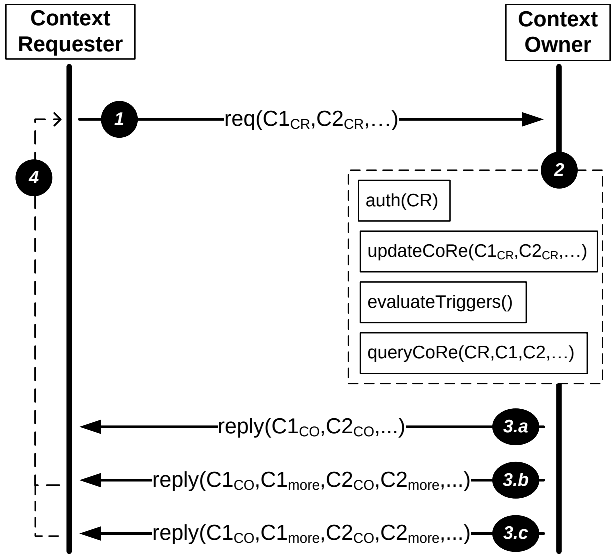


Figure 11. The ALPACA framework's turn-based communication protocol deployed for the exchange of context information between peer users.

the requester belongs to. In case the requesting entity is found to be a fully trusted *local application*, no further checks have to be performed: The PM will look up the *Representation* instances in the CoRe model matching the criteria specified in the request and return them to the context requester. Apart from context-aware applications merely consuming context information, also services performing context reasoning in order to infer high-level context fall in this category. The latter can hence be regarded both as requesters and sources of context information.

If a context requester is a *third-party service*, the PM will query the model to learn if there are any *Grant* instances related to the requester. Hence, if the context owner has set up release triggers that grant the requesting entity access to one or more *Representation* instances, the PM again looks up the most appropriate ones and returns them to the requester. However, if a context requester is not explicitly granted access to any *Representation*, a mediocre representation defined by the user will be returned: To protect a user's privacy, this *Representation* should be highly obfuscated, i.e., of considerably reduced granularity, outright cheated, or simply "*N/A*". Both personal context-aware applications that send data over the Internet as well as multi-subject context aware applications belong to this category. In order to efficiently cater for different needs of context-aware applications, the PM supports request/response-based and publish/subscribe-based communication patterns. Both local and remote requests for a user's context information may hence contain a flag indicating whether the requester is asking for a single reply or continuous updates. In the latter case, a desired update interval as well as a context-dependent break condition can be set by the context requester. As long as there is a corresponding *Grant* instance and the break condition is not yet met, the PM will supply the requester with the requested information.

Due to a user's privacy policies possibly depending on the requesting entity's context as well, the context owner might require her *peer users* to communicate their own context when requesting her context information. Figure 11 outlines the communication flow during a request for context information issued by a peer user. As a means for protecting the requesting entity's privacy, too, we introduce a turn-based protocol for the exchange of context information. Our protocol, which also aims at fulfilling the requirement for symmetry in P2P-based usage scenarios, works as follows: *(1)* The context requester (*CR*) issues a request for some of the context owner's (*CO*) context information. The *CR* implicitly specifies which types and granularities of context information he is interested in by piggybacking equivalent information about himself on the request. *(2)* The *CO*'s PM analyzes the request by authenticating the *CR* and adding the contained context information about the *CR* to the CoRe model. At this step, naturally, the *CO*'s release triggers are re-evaluated based on the updated state of the model. The PM then looks up any *Grant* instances related to the requesting entity based on the model's current state. Additionally, the PM also queries the CoRe model for any active triggers that might match the *CR*, e.g., based on the *CR*'s identity or static group membership. Based on this information, for each context subtype the PM learns if the *CO* is willing to release any *Representation* instances to this requester. The next steps are hence to be performed for each type of context information separately: Depending on whether a corresponding *Grant* instance has been found or not, the PM either replies with the respective *Representation* or with the subtype's highly obfuscated standard representation. *(3.a)* If both no *Grant* instances and no active triggers have been found, the protocol ends here. *(3.b)* Otherwise, if the *CR* is explicitly granted access to a *Representation*, the PM looks up the corresponding instance matching the granularity specified in the request. In case the requester asked for a lower granularity representation than the one that is granted to him by the respective *Grant*, the PM automatically adjusts the granularity by selecting the matching *Representation* using the CoRe model's transitive *generalizationOf* property. Given that the *CO*'s maximum granularity for this situation is not yet reached the PM will notify the *CR* about this. *(3.c)* Additionally, there might be active triggers related to the *CR* that have not fired yet, as they depend on the requester's higher resolution context. If so, the PM also informs the *CR* about the availability of representations of higher granularities that possibly will be released after learning more about the *CR*'s respective context information. In order to reveal as little as possible, however, the *CO*'s PM only indicates the existence of higher resolution information, but does not betray the maximum granularity available. *(4)* Upon reception of the *CO*'s response, the *CR* updates the local CoRe model and decides about entering another round of the protocol in order to learn more details about the *CO*'s context. Notice, that based on the received *CO*'s context the *CR*'s set of active triggers might have beeen updated, too. If so, the *CR* selects a corresponding higher granularity *Representation* in order to issue a new request for the *CO*'s context information and the protocol is repeated until either the *CO* or *CR* reaches a maximum level of granularity and decides to quit.

We hence enforce the symmetry requirement by demanding that the release and resolution of a user's context information directly correlates with the information the requester has to share. Notice, however, that there is no guarantee that each party will always learn just as much about the other one as it conveyed about itself. At each step in the protocol an entity can decide to quit the protocol and not share any further information, possibly depending on the other party's context information just learned. However, the turn-based nature of the protocol tries to alleviate this problem. The communication protocol can be applied to the example from Section VI-C as follows: According to the release triggers that Alice has set up, she only shares her location information with her colleagues during her working hours. If her own location context is known to be *atWork* and so is Bob's, however, she has decided to even let him know in which room she is located. Her mobile device of course knows what time it is and where she is at, yet the current location of Bob must be known as well. Assume that Bob already knows she is at work, so he directly queries her location on a room level. Following the symmetry requirement, he therefore has to tell her the same information about himself. When the request arrives at Alice's PM, Bob's location context can be reasoned to be *atWork*, hence her room-level location information can be released to him. However, in case Bob is not sure about here being at work, he probably would just formulate the request with his own location being reported as *atWork*, which results in Alice's PM stating the same and informing Bob about higher resolution context being available. Notice, that in step *(2)* it is also possible that the requester's context can be used to calculate new local context at the context owner's device, e.g., learning about a peer user's location might lead to a context source being able to detect proximity to that user, which might result in new triggers being added to the context owner's active set.

C. Entity Authentication

Different sources and requesters of context information vary in terms of placement in the operating system and network location, trustworthiness, liability to impersonation attacks, and frequency of requests to the PM. Consequently, adequate means of authentication have to be considered.

Both third-party services and peer users requesting a user's context information from the PM can be authenticated based on public key certificates. Hence, the authenticity of networked requests from these kind of context requesters can be checked by means of digital signatures. In order to protect the user's context information from eavesdropping, the requester's public key can also be used for content encryption. Obviously, the PM only accepts requests for the user's *protected layer* context from entities that are already known. To this end, the PM holds a repository for all those entities' certificates that the context owner is possibly interested in sharing her protected context information with. For identification and authentication of third-party services and multi-subject context-aware applications one can simply rely on a standard public key infrastructure (PKI) and the validity of a given application's certificate, respectively. The exchange of certificates between peer users can be realized by deploying a protocol similar to the process of becoming friends in our privacy-preserving OSN *Vegas* [24]: Users that know each other in real life can perform an out-of-band key exchange using self-signed certificates in order to bypass the need for a PKI. Common difficulties related to certificate revocation can be avoided by these self-signed certificates being valid for a limited time span only and key renewal on a regular basis. Consequently, in order to reliably check the origin and authenticity of an incoming request for context information the PM simply tries to verify the request's digital signature using the information stored in its local certificate repository. Once authenticated, the PM maps the requester to the corresponding entity and queries the CoRe model for related *Grant* instances.

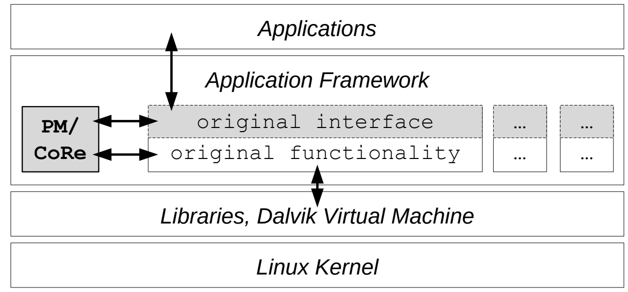


Figure 12. Placement of the *Privacy Manager* in the Android software stack.

Due to its computational overhead, however, the deployment of public key cryptography is overkill for sources and requesters of context information running locally on the user's mobile device and frequently interacting with the PM, e.g., services performing context reasoning. Instead, an HMAC-based approach could be used for lightweight integrity checking and entity authentication. In order to prevent low-level data leakage the problem of securing communication channels between different components on the mobile device itself will have to be further investigated. Initial thoughts on how the PM could blend into a mobile platform's software stack will be presented in the next section.

D. Integration into Android

We will now outline how our framework can be integrated into a modern smartphone software stack such as Android. As shown in Figure 12, the latter is based on a Linux kernel and uses the *Dalvik Virtual Machine* to execute applications written in Java and converted from bytecode to the more lightweight `.dex` (*Dalvik Executable*) format. For security reasons, each application runs in a dedicated sandbox process using its own instance of Dalvik. Apart from standard libraries in C and Java, Android ships with an *Application Framework* that can be used by developers to access a mobile device's base functionality, such as telephony and window management, and also to request context information, e.g., the current GPS position fix. With the PM working as a gatekeeper responsible for handling all requests for updating and querying the user's context information, this layer of the Android software stack is where our framework's key component is to be placed.

In order to effectively protect a user's context information, applications must no longer be able to access the original functionality of the *Application Framework* directly. Applications must yet still be able to make usage of the well-established programming interfaces provided by the Android SDK in order to request desired information. Consequently, even the existence of the PM should be transparent to applications. This can be achieved by exchanging the standard implementation of all the respective interfaces for proxy code, which redirects corresponding requests to the PM along with information about the context requester's identity. Based on the current context and the privacy layer the requester belongs to, the PM then can decide which kind of information is to be released to the requesting entity. Apart from this kind of legacy functionality, however, applications and services might be especially designed to work with the ALPACA framework, e.g., peer-based context-aware applications, which directly register with the PM in order to increase performance by calling optimized interfaces. For context acquisition the PM can simply run the original interfaces' code for infering the user's current context, feeding this information into CoRe, and updating the set of active release triggers. Obviously, also different kinds of covert channels that can be used by applications to collect information without explicitly requesting it will have to be identified and eliminated. In addition, the PM must also be able to monitor whether applications behave as expected, i.e., act in accordance to the corresponding privacy layer. For example, context requesters must not be able to trick a user into believing that an application is to be placed on the *private layer* while still sending data off the device. Hence, the PM must not only be in a position enabling it to manage access to context information, but must also be able to control what happens to this information afterwards. To this end, we are currently investigating the applicability of approaches for information-flow tracking, such as *TaintDroid* [25], and plan to extend ALPACA with the necessary functionality.

VIII. Discussion

Strictly following a privacy-centric point of view, all decisions in the design process have been directed at offering users a maximum level of control over the release of context information. Hence, we will now check our framework against the requirements stated in Section II and dicuss the pros and cons of our privacy-centric approach for modeling and managing a user's context information.

In a first step, this article introduced an abstract conceptualization of privacy needs regarding context information. To this end, four different privacy layers and appropriate inter-layer gatekeeping mechanisms have been identified, which render a user's privacy preferences to be effectively manageable by a software system. These layers provide a good compromise about information accessibility, flexibility, and reduction of complexity: A blacklist-based approach can be used to control what kind of context information should be acquired by the user's device. For the release of context information, however, a whitelist-based approach is pursued, argueing that a privacy-aware user wants to feel rather safe than sorry. Consequently, each access to context information has to be explicitly granted by the context owner. The next goal was to find a formal representation of a user's context information that inherently features general aspects of privacy and is easily extensible with regard to the integration of novel privacy and security methods. Based on an abstraction of context in terms of semantically different representations of the same type of information, we designed the ontology-based CoRe model. The latter is able to meaningfully store multiple versions of a user's context in parallel, which can be used to serve different responses to distinct groups of variably trustworthy context requesters. In addition, we presented an effective mechanism for controlling the dissemination of context information in form of user-defined privacy policies. For this purpose, the ALPACA framework utilizes a two-stage rule-based approach for the definition of fine-grained and context-dependent release triggers on behalf of the context owner. The triggers' conditions, effects, and affected context requesters can be specified based on the rich context information modeled in CoRe, which allows for great expressivity and flexibility. Consistency of a user's privacy policies can be detected at runtime by means of ontology-based reasoning on our model's current state. However, using ontology-based reasoning on a per-request basis might become a performance issue once the information modeled in CoRe becomes extensive, which has to be investigated in our future work. Finally, we introduced the *Privacy Manager* as our framework's key component and described the interplay of the different components, which have been integrated into a system architecture suitable for all kinds of context requesters, e.g., ranging from local applications to peer-to-peer and third-party services. By omitting a TTP, there is no single point of trust or failure. Instead, each user keeps the complete control over the release of her context information. Multi-subject context-aware applications can still be realized, yet in each situation each application is only able to learn the amount of information the context owner is willing to release.

In order to offer maximum levels of universality and extensibility, we intentionally decoupled our framework from the acquisition of context information. It is hence independent from existing sensors, reasoning mechanisms, and context obfuscation techniques implemented on the user's device. The latter can be used to realize the concepts of variable granularity, plausible deniability, etc. as required. New sensors and inference algorithms can simply register themselves at the PM without requiring the need for modifications to any of the ALPACA components. In a practical implementation, the PM can simply inform the user about new types or sources of context information being available.

Due to its restrictive nature, however, our approach inevitably comes at the expense of usability and out-of-the-box functionality: All context information the context owner wants to be accessible by other entities in certain situations have to be explicitly released by manually setting up appropriate release triggers. It is thus not clear whether a majority of users is willing to adopt such a whitelist-based system, e.g., facing peer pressure and their own reluctance towards manually configuring the release of context information. The deployment of obtrusive notification mechanisms is also likely to be perceived as being disturbing, such as alerting the context owner about the release of context information on a per-request basis. However, we argue that there is a clear necessity to give users full control over the release and granularities of their context information even if usability is constrained. Moreover, the general privacy awareness of mobile users has to be trained in order to prevent unnoticed large scale data leakage. Nevertheless, user-friendly mechanisms will have to be developed in order to increase the likeliness of a broad acceptance among a wide user base. For example, a layered policy approach as proposed in [17] could be applied, which allows both for simple privacy settings suitable for the mostly unconcerned and pragmatic users and fine-grained policy-definitions for fulfilling the needs of the privacy-aware. Furthermore, it has been shown in [16] that a personalized recommendation of most likely privacy policies based on a user's current context, previously learned preferences and group correlation is feasible, too, which presents another interesting direction for future research.

IX. Conclusion

This article presented ALPACA, an integrated framework for modeling and managing a user's context-information in a privacy-centric way. To this end, we introduced a practical conceptualization of privacy in context-aware applications with regard to who is requesting information from the user's mobile device. On the basis of this abstracted view we presented our ontology-based CoRe model, which can be used for maintaining multiple, semantically different representations of the same class of context information fit for differently trustworthy groups of context requesters. In order to enable the fine-grained definition of a user's privacy preferences a context-dependent, whitelist-based trigger mechanism has been created. Eventually, we described our framework's key components as well as its system architecture and a turn-based communication protocol used for the exchange of context information between different entities.

We are currently working on a prototype implementation of our system allowing us to conduct a user study for evaluating the usability of our whitelist-based release mechanism. As for our future work, we aim at finding mechanisms capable of ensuring consistency over several consecutive context requests by a single entity. Furthermore, we are looking for a way to implement incentive-based mechanisms for the optional release of additional data to third-party services in a privacy-preserving way. Future work will also be directed at finding new obfuscation mechanisms for different types of context information and integrate them into ALPACA. Finally, we want to investigate techniques increasing usability, such as an automatic mapping of applications to the privacy layers of our framework.

References

[1] F. Dorfmeister, S. Feld, C. Linnhoff-Popien, and S. Verclas, "Privacy-centric modeling and management of context information," in CENTRIC 2013, The Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services, 2013, pp. 92–97.

[2] A. Amir, A. Efrat, J. Myllymaki, L. Palaniappan, and K. Wampler, "Buddy tracking - efficient proximity detection among mobile friends," Pervasive and Mobile Computing, vol. 3, no. 5, pp. 489–511, 2007.

[3] A. Küpper and G. Treu, "Efficient proximity and separation detection among mobile targets for supporting location-based community services," SIGMOBILE Mob. Comput. Commun. Rev., vol. 10, no. 3, pp. 1–12, Jul. 2006, last accessed on 2014-06-16. [Online]. Available: http://doi.acm.org/10.1145/1148094.1148096

[4] F. Dorfmeister, M. Maier, M. Schönfeld, and S. Verclas, "Smartbees: Enabling smart business environments based on location information and sensor networks," in 9. GI/KuVS-Fachgespräch "Ortsbezogene Anwendungen und Dienste". Springer, 2012, pp. 23–37.

[5] M. Ackerman, T. Darrell, and D. J. Weitzner, "Privacy in context," Human–Computer Interaction, vol. 16, no. 2-4, pp. 167–176, 2001.

[6] G. D. Abowd et al., "Towards a better understanding of context and context-awareness," in Handheld and ubiquitous computing. Springer, 1999, pp. 304–307.

[7] P. Makris, D. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: On networking and computing environments' integration," Communications Surveys Tutorials, IEEE, vol. 15, no. 1, pp. 362–386, First 2013.

[8] A. Solanas, J. Domingo-Ferrer, and A. Martínez-Ballesté, "Location privacy in location-based services: Beyond ttp-based schemes," in Proceedings of the 1st International Workshop on Privacy in Location-Based Applications (PILBA), 2008, pp. 12–23.

[9] V. Sacramento, M. Endler, and F. N. Nascimento, "A privacy service for context-aware mobile computing," in Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on. IEEE, 2005, pp. 182–193.

[10] M. Blount et al., "Privacy engine for context-aware enterprise application services," in Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on, vol. 2. IEEE, 2008, pp. 94–100.

[11] H. Chen, T. Finin, and A. Joshi, "An intelligent broker for context-aware systems," in Adjunct proceedings of Ubicomp, vol. 3, 2003, pp. 183–184.

[12] L. Kagal, T. Finin, and A. Joshi, "A policy language for a pervasive computing environment," in Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on. IEEE, 2003, pp. 63–74.

[13] R. Wishart, K. Henricksen, and J. Indulska, "Context privacy and obfuscation supported by dynamic context source discovery and processing in a context management system," in Ubiquitous Intelligence and Computing. Springer, 2007, pp. 929–940.

[14] W. Apolinarski, M. Handte, D. Le Phuoc, and P. J. Marrón, "A peer-based approach to privacy-preserving context management," in Proceedings of the 7th international and interdisciplinary conference on Modeling and using context, ser. CONTEXT'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 18–25, last accessed on 2014-06-16. [Online]. Available: http://dl.acm.org/citation.cfm?id=2045502.2045505

[15] A. Behrooz and A. Devlic, "A context-aware privacy policy language for controlling access to context information of mobile users," in Security and Privacy in Mobile Information and Communication Systems, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, R. Prasad et al., Eds. Springer Berlin Heidelberg, 2012, vol. 94, pp. 25–39.

[16] J. Xie, B. P. Knijnenburg, and H. Jin, "Location sharing privacy preference: Analysis and personalized recommendation," in Proceedings of the 19th International Conference on Intelligent User Interfaces, ser. IUI '14. New York, NY, USA: ACM, 2014, pp. 189–198, last accessed on 2014-06-16. [Online]. Available: http://doi.acm.org/10.1145/2557500.2557504

[17] W. Bokhove, B. Hulsebosch, B. Van Schoonhoven, M. Sappelli, and K. Wouters, "User privacy in applications for well-being and well-working," in AMBIENT 2012, The Second International Conference on Ambient Computing, Applications, Services and Technologies, 2012, pp. 53–59.

[18] K. Sheikh, M. Wegdam, and M. Van Sinderen, "Quality-of-context and its use for protecting privacy in context aware systems." Journal of Software (1796217X), vol. 3, no. 2, 2008.

[19] A. Kofod-Petersen et al., "Implementing privacy as symmetry in location-aware systems," in Proceedings of the International Workshop on Combining Context with Trust, Privacy and Security (CAT 2008), G. Lenzini, B. Hulsebosch, S. Toivonen, and J.-M. Seigneur, Eds, vol. 371, 2008, pp. 1–10.

[20] L. Sweeney, "k-anonymity: A model for protecting privacy," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 05, pp. 557–570, 2002.

[21] C. Bettini et al., "A survey of context modelling and reasoning techniques," Pervasive and Mobile Computing, vol. 6, no. 2, pp. 161–180, 2010.

[22] R. Reichle et al., "A comprehensive context modeling framework for pervasive computing systems," in Proceedings of the 8th IFIP WG 6.1 international conference on Distributed applications and interoperable systems, ser. DAIS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 281–295, last accessed on 2014-06-16. [Online]. Available: http://dl.acm.org/citation.cfm?id=1789074.1789105

[23] T. Strang, C. Linnhoff-Popien, and K. Frank, "Cool: A context ontology language to enable contextual interoperability," in Distributed applications and interoperable systems. Springer, 2003, pp. 236–247.

[24] M. Dürr, M. Maier, and F. Dorfmeister, "Vegas – a secure and privacy-preserving peer-to-peer online social network," in Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom), Sept 2012, pp. 868–874.

[25] W. Enck et al., “Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones,” in Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, ser. OSDI’10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–6, last accessed on 2014-06-16. [Online]. Available: http://dl.acm.org/citation.cfm?id=1924943.1924971

Geo-Temporal Visual Analysis of Customer Feedback Data Based on Self-Organizing Sentiment Maps

Halldór Janetzko, Dominik Jäckle and Tobias Schreck
University of Konstanz
Konstanz, Germany
Email: Halldor.Janetzko@uni.kn, Dominik.Jaeckle@uni.kn, Tobias.Schreck@uni.kn

Abstract—**The success of a company is often dependent on the quality of their Customer Relationship Management (CRM). Knowledge about customer's concerns and needs can be a huge advantage over competitors but is hard to gain. Large amounts of textual feedback from customers via surveys or emails has to be manually processed, condensed, and lead to decision makers. As this process is quite expensive and error-prone, CRM data is in practice often neglected. We therefore propose an automatic analysis and visualization approach helping analysts in finding interesting patterns. We combine opinion mining with the geospatial location of a review to enable a context-aware analysis of the CRM data. Instead of overwhelming the user by showing the details first, we visually group similar patterns together and aggregate them by applying Self-Organizing Maps in an interactive analysis application. We extend this approach by integrating temporal and seasonal analyses showing these influences on the CRM data. Our technique is able to cope with unstructured customer feedback data and shows location dependencies of significant terms and sentiments. The capabilities of our approach are shown in a case-study using real-world customer feedback data exploring and describing interesting findings.**

Keywords-customer relationship management, review analysis, self-organizing maps, sentiment analysis.

I. Introduction

Many companies with business in the world wide web collect reviews and customer feedback of their products and services. One common way of assessing customer satisfaction are grading schemes (e.g., one to five stars) and free text forms allowing more detailed customer comments. But aside from showing the average rating or the distribution of ratings, more sophisticated and consequently also more expressive analyses are performed very rarely. This is surprising, as the free text provided by customers is a valuable source of hints with respect to customer needs and satisfaction levels, but a manual inspection is often not feasible. Modern approaches of text processing and visualization can help at this end, by summarizing important themes and sentiments in large amounts of text.

An effective analysis of textual customer feedback should involve and examine different aspects of the text content. The most obvious one is the *frequency of statements or terms*. Simple statistics and visualization methods like word clouds may help to get a first impression of most important keywords. But simple statistics do not help to analyze, whether the customers liked or disliked these points. The next important aspect is the *sentiment* extracted from the context of the addressed keywords occurring in the text. E.g., customers may complain or praise products or services, and by using sentiment analysis, we aim at capturing this notion. From a company's point of view, negative statements are in many cases more important to analyze than the positive ones, to improve customer satisfaction. But the computation of one single sentiment score is not very expressive as customers might review more than one aspect, and different customers may have different opinions. Therefore, the challenge is to arrive at a fine-grained analysis of this complex data. The sentiment analysis should assign sentiment scores with respect to the attributes of the product or service, instead of computing one value. Customers, for instance, could like a certain bought product, at the same time complain about a too complicated ordering process.

Yet another key aspect holding valuable information in customer feedback data is the *geospatial location*. Customer feedback can be geolocated by several ways, including having the customer address in a corporate database, or by geo-resolving the IP address an anonymous web feedback was provided. From that we can derive the geospatial distribution of customer feedback, which is important for two reasons. First, for global companies, cultural differences may influence the customers' conception and country specific products or services should be offered. Second, besides cultural differences there is another aspect which may change customer's needs. The geographic location determines for instance the climate and may also impose delivery obstacles resulting from the geographic topology. In very dry areas, for example, it may be reasonable to leave a parcel outside the customer's housing, but in rainy areas the customer may complain about a soaked product. Concerning the topology, hard to reach customers (e.g., islands or exclaves) may complain about long delivery times, but there may be nothing the company could do about it.

This paper is an extended version of our previous work [1]. Compared to the previous work, we additionally take the time dimension of the customer feedback data into account. Performing temporal and seasonal analyses reveals further interesting patterns and insights.

Our main motivation for this work was the following starting hypothesis, to be explored on a real-world CRM data set: "The geographic position of reviewing customers correlates to

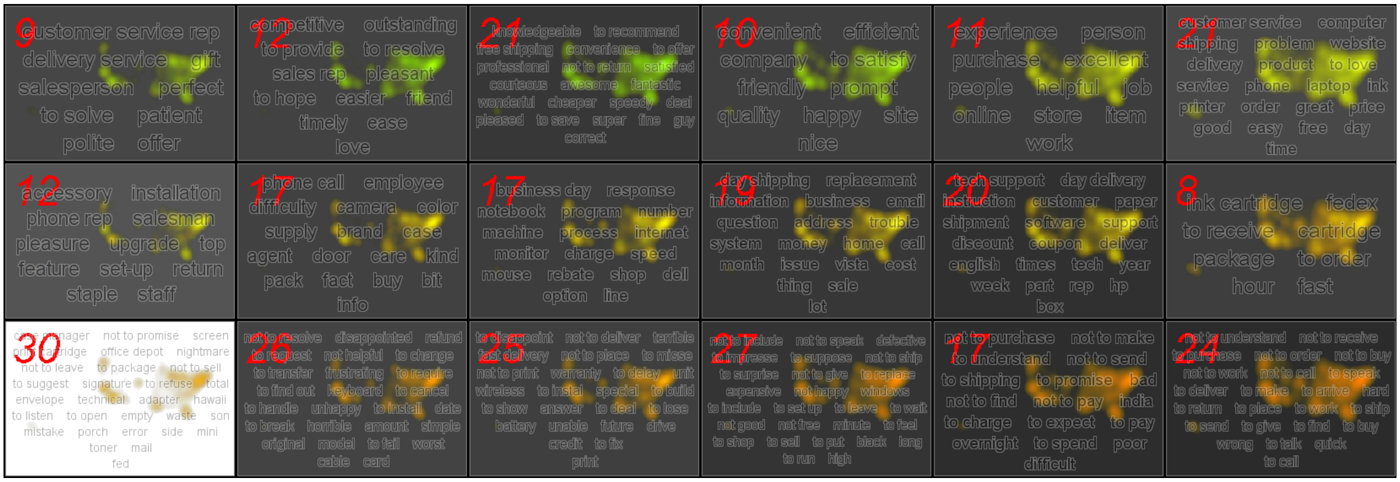


Figure 1. Visualization of customer feedback data using sentiment analysis and self organizing maps. Term groups with different sentiment scores are visible and additionally the geospatial distribution of the terms is displayed. The brightness of the background represents the coherence of the geographic distribution for each SOM node. [1]

their satisfaction levels and needs." We wanted to see, whether there are differences in customer preferences caused by the geospatial location. The result of this analysis could help to improve the customer satisfaction by detecting differences in customer needs. Companies can therefore differentiate better among their customers and can easily focus and channel their efforts. Furthermore, we believe that the season of the review influences the sentiment for some terms. It is interesting to see which terms are affected by the temporal dimension and which not, to improve the company's reactions. For instance, the need for support could be much higher after Christmas, as typically many technical devices are bought in this time period.

In this paper, we perform customer feedback analysis based on *sentiment maps*. Sentiment maps are the result of preceding opinion mining steps, where the occurrence of a term is drawn on a geographic map. The color used hereby depicts the sentiment and the sentiment map consequently shows not only the geospatial distribution of the term but simultaneously also the sentiment distribution. Following this approach leads to one sentiment map for each term. Further details of this approach can be found in the beginning of the analysis results section of this paper. A result of our technique is depicted in Figure 1.

We present in this paper our methodology analyzing customer feedback with respect to sentiment and geospatial customer location. Our contributions are the combined text and geospatial analysis of customer feedback data and the visual representation allowing a comparative analysis. Furthermore, we show that there are indeed frequent feedback terms (concepts) with a high geospatial dependency. The paper is structured as follows. First, we will give an overview to existing and related work in Section II, and then detail our approach in Section III. Findings from an application to a real-world data set will be discussed in Section IV. We will conclude with an outlook to future work.

II. Related Work

Our work relates to the wider area of visual data analysis. We discuss a body of work on visual cluster analysis with Self-Organizing Maps, on analysis of geospatial and time-oriented data, and on feature-based text visualization.

A. Self-Organizing Maps for Visual Data Analysis

Many problems in visual data analysis require the reduction of data to perform meaningful analysis on a reduced version of data. Clustering reduces the data to a smaller number of groups to easier analyze and compare; and dimensionality reduction reduces the number of dimensions of data items to consider, and to project data to 2D displays. The Self-Organizing Map (SOM) algorithm [2] is a well-known method, which provides both data reduction and projection in an integrated framework. As a neural-network type method it learns a set of prototype vectors arranged on a regular grid, typically embedded in 2D. The method typically provides robust results in both data clustering and 2D layouting. Using regular 2D grids as neural structures for the SOM training, visualization in form of heatmaps, component planes, and distance distributions comprise basic methods for visual exploration of data using SOM processing [3]. SOM-based Visual Analysis to date has considered different application domains, including financial data analysis based on multivariate data models [4], analysis of web clickstream data using Markov Chain models [5], trajectory-oriented data [6], or time-oriented data [7]. Image Sorter [8] proposed to visually analyze collections of images by training a SOM over color features extracted from the images. We here follow that idea, in that we analyze geospatial heatmaps of sentiment scores using SOM of respective color features as well.

B. SOM-Based Visual Analysis of Geospatial Data

Many application problems involve georeferenced data items, and visual analysis approaches have been identified as very helpful also for geospatial data analysis processes [9]. Choropleth (or thematic) maps are a basic, popular technique to show the distribution of a scalar value over a land-covering map [10]. Also, SOM-based approaches have been studied in context of geospatial data analysis, and proven useful to this end. When considering georeferenced data with SOM, basically two approaches exist. First, in the *joint data model*, one single data representation is formed by combining spatial and other multivariate data into a single vector representation which is input to the SOM method. Examples include [11], where a joint vector representation for both geolocation and demographic data was formed for census data analysis. More methods can be found in [12]. As a second approach, *linked views* integrate visual data analysis of each data aspect (geolocation, time, multivariate measures, etc.) in separate views combined by Brushing & Linking. One example system is [13], where a linked view system proposed the joint visual analysis of geospatial and multivariate data. Also, in [14], we proposed to jointly analyze geospatial and temporal phenomena by a linked view. There, SOM clusters can be computed for either data perspective, and the correspondence of clusters to the other perspective is shown by an auxiliary view. In our approach we do not consider geolocation data explicit for the SOM generation, but implicitly by the spatial-sensitive color features extracted from sentiment heatmaps generated from text data (cf. also Section III for details).

C. Visual Analysis of Temporal Data

Also, the temporal dimension plays an important role in many data analysis problems. Typical tasks relating to analysis of time include the comparison of data across time, finding similarities, trends, and correlations among variables. The time dimension can be accommodated for analysis either interactively or automatically. On the interactive side, users may select time intervals of interest for navigation and drill-down operations. On the automatic side, many approaches exist for detecting interesting intervals in time series [15], or finding clusters of similar temporal patterns for seasonal comparison [16]. There exists a rich body of work on visualization of time series data. Specific visual representations have been proposed to cope with the many challenges of time-oriented data, including long, multivatiate, unevenly spaced or categorical time series [17]. In one previous work, we proposed an interactive system for cluster-based analysis of time series data sets, which also included the possiblity to search by visual example, e.g., allowing the user to sketch a target line chart to retrieve [18]. In that system, the SOM approach was found useful to support the exploration of large time series data. While the SOM method can be an effective layout generator to compare many time series, there exist alternative layout schemes which can rely on user-selected or content-based ordering schemes to create small multiple displays. In [19], a taxonomy of layout strategies is given. In this paper, we rely on the SOM for generating geospatial and text-oriented data displays for comparison of customer feedback data. We will make use of small multiple displays to compare situations across time for identification of trends and patterns.

D. Feature-based Text Visualization

Finally, we relate to a body of work in visual document analysis. In general, *feature-based document analysis* abstracts a document (or collection, or stream) by a set of features which are more easy to visualize, as compared to the content of the documents. Numerous document features for different applications have been studied to date. For example, features scoring the readability of documents have been proposed in [20], and features applicable to classify authorship of documents have been surveyed in [21]. Sentiment features rate the polarity (in terms of positiveness of negativeness of statements) in a given text. In combination with time-series analysis, sentiment features can be used, e.g., to detect critical customer opinions in near real time, as possibly arising from some feedback channel [22]. In [23], we applied sentiment analysis to customer feedback data and analyzed it by means of geospatial heatmaps generated for the sentiments. While in [23], we considered only small sets of such heatmaps which we sequentially inspected, the focus of our work here is the comparative analysis of large numbers of sentiment maps, based on the SOM method.

III. TECHNIQUE

Our approach enables the geospatial visual comparison of customer feedback sentiments by using a Self-Organizing overview display. Figure 2 shows the overall process that is divided into four steps: (1) First, we extract a color feature vector for each sentiment map. (2) Second, we train the SOM and assign every sentiment map exactly one node. In step (3), we aggregate all sentiment maps that are located on the same map node. (4) Finally, we calculate the coherence and enhance the aggregated sentiment map with the content terms from the represented customer review texts. To analyze temporal data, we partition the dataset according to the date and process the proposed pipeline multiple times for each partition separately. We afterwards use of small multiple displays to compare different dates. We next detail these steps.

A. Process Pipeline

In order to process one partition of a temporal varying dataset, the introduced pipeline is processed exactly once. In this section, we describe the four different steps towards a sentiment SOM, before treating multiple temporal varying datasets.

Feature Vector Extraction. The feature vector we use as input to the SOM computation consists of localized RGB color values. We create a grid overlay for each sentiment map and calculate the color mean value for each cell. The mean value is determined by the color value of each single pixel contained in the corresponding grid cell. The representative feature vector for any sentiment map is created using the RGB color model. All RGB mean values are forming the feature vector:

$$\text{Picture1} = (R_{1,1}\, G_{1,1}\, B_{1,1}\, R_{1,2}\, G_{1,2}\, B_{1,2}\, R_{1,3}\, G_{1,3}\, B_{1,3} \ldots)$$

$R_{i,j}$ represents the value of R for picture i and cell j. This format is used as feature vector representing one sentiment

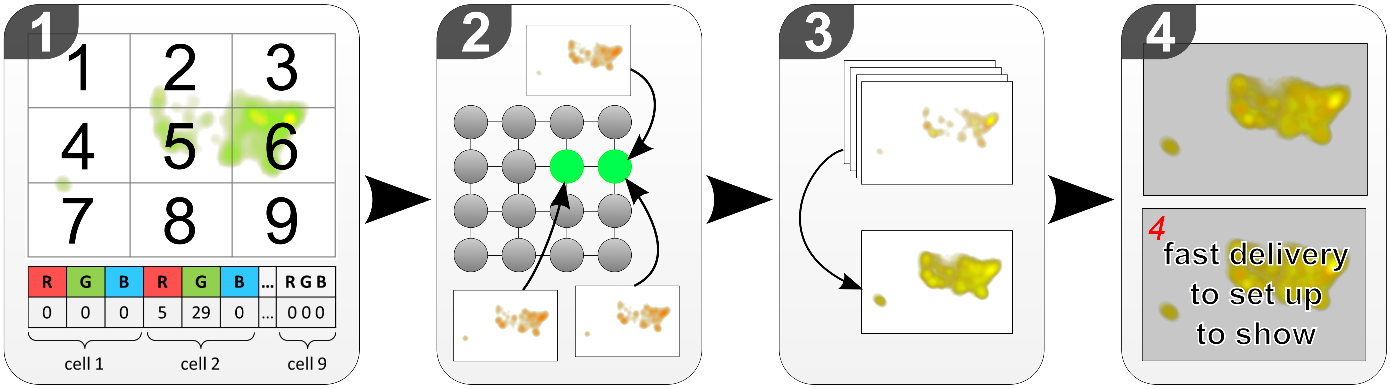


Figure 2. Process pipeline to create self-organizing sentiment maps. The self-organizing sentiment map is determined by (1) extracting corresponding feature vectors, (2) calculating the best matching unit, (3) aggregating similar images according to the SOM, and finally (4) mapping the coherence and the additional information like the used terms to the images. [1]

map; each picture is assigned exactly one vector. Then, the extracted feature vectors are used to train the SOM using the SOMPAK implementation [24] (see also Figure 2 (1)).

Sentiment Map Classification. We apply a standard SOM training process following best practices suggested in [24]. Based on the defined SOM grid resolution, the prototype vectors are linearly initialized. Then, two learning phases are applied. First, a coarse learning is performed with a larger training radius, so that every considered node has a wide impact factor. Then, a fine-tuning training step is performed with a smaller training radius. Once the SOM-training has finished, the best matching prototype vector on the SOM grid (best matching unit, or bmu) is determined for each sentiment map by finding the node with the minimum distance (1).

$$bmu(SM) = \min_{k=1}^{M} \left(\sqrt{\sum_{i=1}^{N} (v(SM).i - v(node_k).i)^2} \right) \quad (1)$$

We iterate over all sentiment maps and calculate the best matching unit for each sentiment map SM. Then, we iterate all M trained SOM nodes and calculate the minimum Euclidean distance between the sentiment map and the trained SOM node. Therefore, the feature vector of the sentiment map and the vector of the SOM node are used. The corresponding vector is determined via the function $v()$ with size N. The control variable i addresses every single vector entry. Finally, the sentiment map is assigned to the SOM node with the minimum distance (see also Figure 2 (2)). The grid size can be chosen individually for each application.

Similarity-based Sentiment Map Aggregation. As the outcome of the SOM and bmu mapping, multiple sentiment images may share the same SOM node. Therefore, we need to provide aggregation of such sets of maps. To find a representative image for those sentiment maps different approaches are possible. We here chose to apply visual aggregation and merge all similar images into one. Therefore, every sentiment map is assigned a transparency value, so that we are able to create one image by lying one sentiment map upon each other. The resulting image visualizes all aggregated sentiment areas. By adding multiple pictures on top of each other, the last added picture on top has the highest impact according to the process of alpha composition in terms of occlusion [25]. For that reason, we calculate the intersection of sentiment maps on our own based on the color, shown in Figure 2 (3).

Coherence Mapping and Map Enhancement. The last step of the pipeline is twofold: First, we map the background of the aggregated sentiment map to its coherence. Second, we enhance the aggregated sentiment map with additional information.

Aggregating multiple sentiment maps may result in an image showing a constant distribution. But when comparing all contained sentiment maps, the sentiment maps might be very diverse regarding to geospatial distributions. In order to understand the composition of those aggregated sentiment clusters it is important to define a quality criterion: the coherence of the sentiment maps. Thus, we make use of the background and define a coherence measure. The coherence measure (2) expresses how similar two sentiment maps are according to its feature vector. The coherence is mapped to the color range from black (high coherence) to white (very low coherence).

$$coherence(SMS) = \frac{\sum_{i=1}^{N} \sum_{j=i}^{N} dist(SMS_i, SMS_j)}{\frac{N \cdot (N+1)}{2}} \quad (2)$$

The $coherence$ is calculated for N sentiment maps (SMS) addressing the same SOM node. Summarizing, we build the average of all pictures including a distance function. We then sum the distance value of each sentiment map to all other sentiment maps. The distance function $dist$ between two sentiment maps is defined in equation (3).

$$dist(p, q) = \frac{\sqrt{\sum_{k=1}^{M} (v(p).k - v(q).k)^2}}{|\{i \in 1..M : \neg(v(p).i = v(q).i = 0)\}|} \quad (3)$$

The distance function requires two sentiment maps as parameters with dimension M. The Euclidean distance is normalized by the number of vector dimensions M excluding all dimensions with zero values in both dimensions. The problem of a possible low coherence raises with the algorithm of the SOM: To calculate the similarity in the second step the Euclidean distance combines sentiment maps that are very sparse, as the exact locations do not matter.

Sentiment Keyword Visualization. Every sentiment map corresponds to one term. As a consequence, if multiple sentiment maps are aggregated, the resulting image corresponds to multiple terms. Hence, we combine the aggregated sentiment maps with a simple but effective text representation: All terms are drawn semi-transparent with a gray border on top of the aggregated image. Also, the amount of sentiment maps that have been aggregated is indicated by a red number on the top left corner. Using an intelligent text layout algorithm, the analyst can easily identify the terms corresponding to the image; the text uses the full width and height to be easy to read. Figure 2 (4) illustrates the automatic labeling result.

Depending on the chosen grid size in the first step (feature vector extraction), the final result may differ. To allow data abstraction and overview large data sets, we typically chose a relatively small grid size, where the amounts of nodes is significantly smaller than the amount of considered sentiment maps.

B. Comparable Analysis of Temporal Data

For the analysis of temporal data, our technique needs to be extended as shown in Figure 3. In the previous section, the data according to only one time window has been processed in order to create the sentiment map. In case of a temporal dataset, we adapt our proposed technique: First, the temporal data is partitioned by date respectively. Second, the process pipeline is executed for each data partition separately; but the training phases change to obtain comparable layouts. Finally, all resulting sentiment maps are layouted and visualized. The visualization allows to compare multiple datasets by using the technique of small multiples.

Data Partitioning. To analyze the temporal dimension, we partition the data corresponding to predefined intervals by the user. For the analysis of seasonal trends, for example, the dataset is divided into the quarters of a year or years respectively. Each partition is treated separately.

Small Multiple Displays is a grid-like visualization technique allowing the comparison of similar graphics. We make use of this technique to visualize multiple sentiment map instances at the same time; this enables a seasonal comparison. We chose a square algorithm to layout all small multiples via a grid. Each grid cell is assigned one sentiment map. Therefore, we initially calculate the grid before assigning the sentiment maps.

$$rows = \left\lceil \sqrt{|SM|} \right\rceil \tag{4}$$

$$cols = \left\lceil \frac{|SM|}{rows} \right\rceil \tag{5}$$

We calculate the amount of rows ($rows$) by ceiling the result of the square root of the amount of all sentiment maps SM.

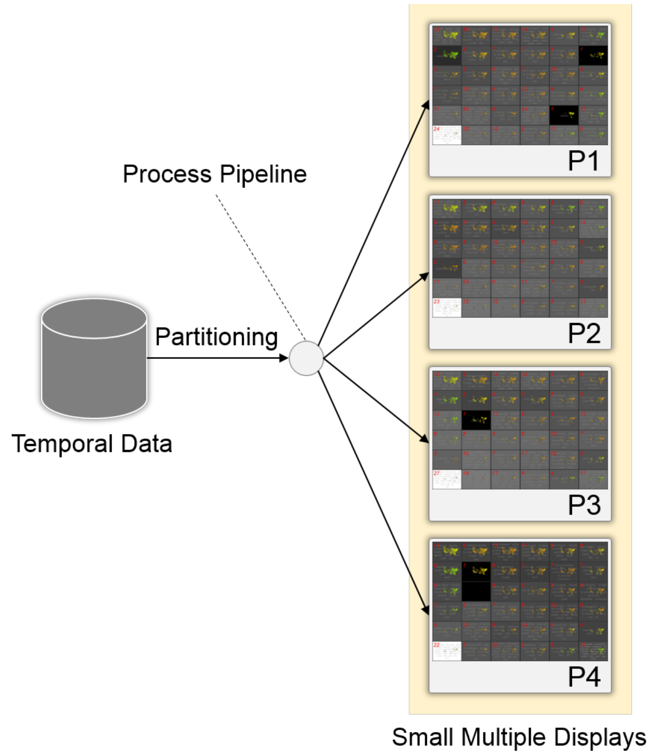


Figure 3. Adapted pipeline towards seasonal sentiment maps. First, the temporal data is partitioned according to user selected temporal intervals. Second, the process pipeline (see Figure 2) is processed for each partition separately in order to create small multiple displays. The sentiment maps are made comparable by performing the fine training on the same feature vector basis.

Then, we use this value to calculate the amount of columns $cols$. This step is especially necessary for an odd amount of sentiment map results. After the necessary amount of rows and columns has been calculated, a uniform width and height is assigned to each grid cell. In order to allocate resulting sentiment maps, we iterate through all columns and all rows assigning the corresponding spot to the sentiment map.

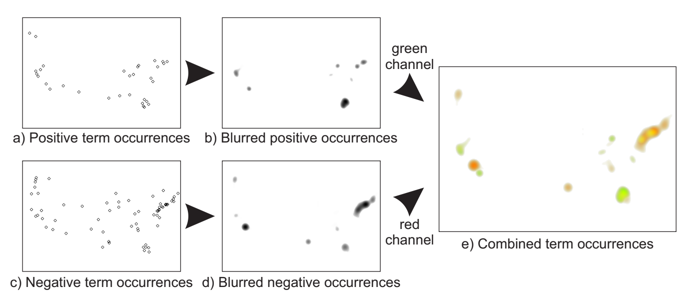


Figure 4. The location maps of positive (a) and negative (c) review terms are blurred to increase the visual saliency and to give a visual aggregation (b, d). Both blurred location maps are then combined (e) by using the RGB channels and show the distribution of positive (green), neutral (yellow) and negative (red) term occurrences. [1]

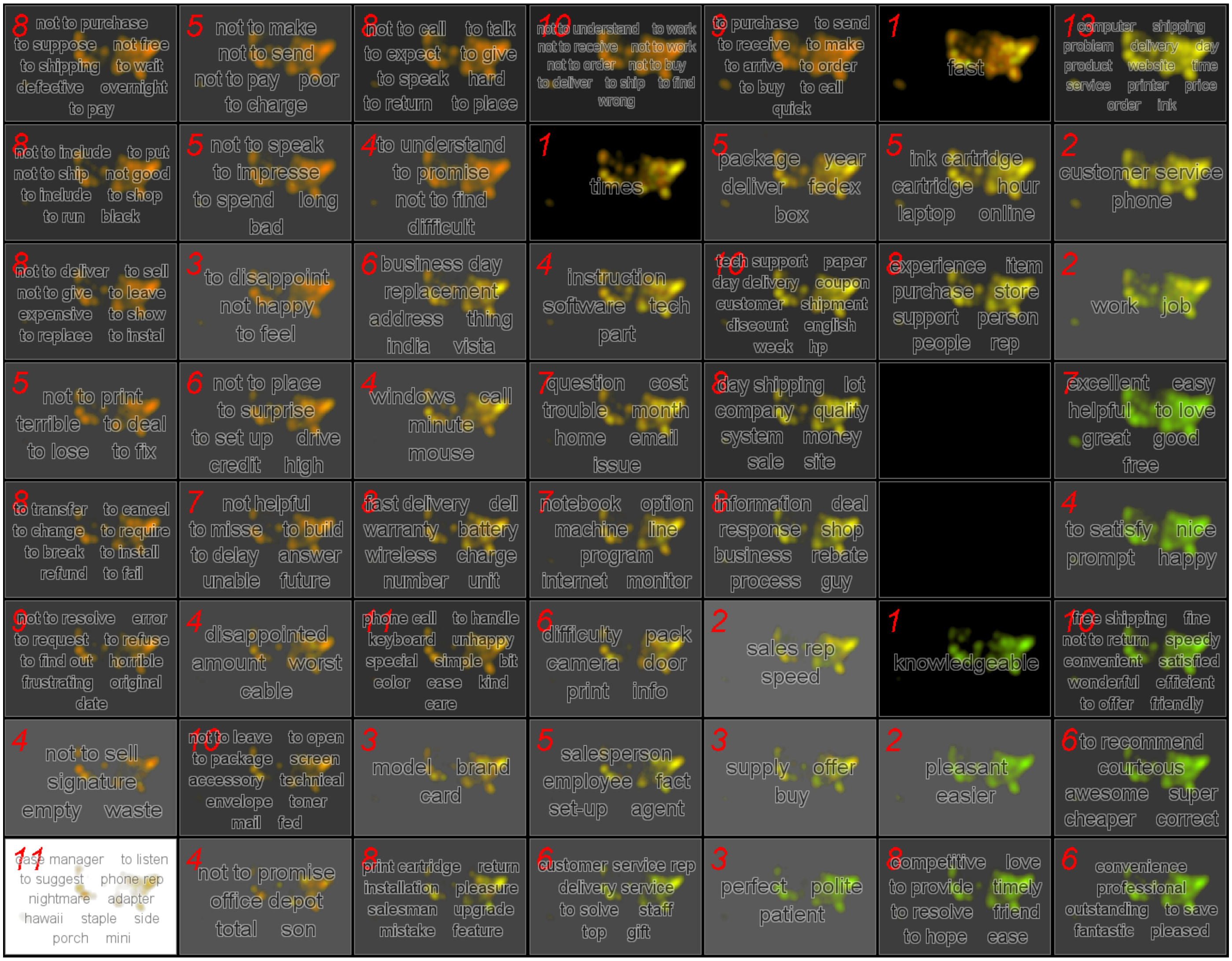


Figure 5. Resulting overview visualization of the SOM analysis process. There are SOM nodes like the one in the lower left being highlighted with a bright background showing a high internal diversity. [1]

IV. Analysis Results

We applied our methods described above to sentiment maps of a real-world data set of collected customer reviews. The reviews were collected after online purchases via an online survey. The data set consists out of 86,812 customer reviews with an average of 18.4 words per review (the median is 12 words per review). In this section, we will first describe the input images resulting from a technique called sentiment maps more in detail. Afterwards, we will discuss interesting findings with respect to the geographic distribution of frequently reported review terms.

Sentiment Maps. Sentiment maps allow the user to inspect the geospatial sentiment distribution of individual terms and are introduced in [23]. After collecting all terms of all reviews excluding stop words one visualization for each of these terms is created. More specifically, first all occurrences of the respective term are determined and the sentiment value for these occurrences are retrieved. The data is then used to generate the sentiment map as illustrated in Figure 4. The data is first partitioned into two subsets: the occurrences with positive sentiment in Figure 4(a) and occurrences with negative sentiment in Figure 4(c). The two partitions are processed separately. A Gaussian blurring function is applied in order to spatially extend the occurrences and increase the visual salience of the geospatial distribution patterns. The result is a blurred representation for both sentiments showing the respective geospatial occurrences as depicted in Figures 4(b) and 4(d). Finally, a combined image is created by using the RGB channels of the RGB color model. The blurred image of the negative occurrences is put in the red channel, and the green channel is used for the positive occurrences. Consequently, locations with both positive and negative sentiments will result in yellow colors, while pure positive sentiments will result in green colors. We did not differentiate within negative sentiments or positive sentiments respectively as this differentiation is highly user and application dependent. But sentiment maps could be extended by this possibility. The final result of this technique can be seen in Figure 4(e).

Geospatial Analysis and Findings. We applied the technique described in Section III to a dataset consisting of 327 sentiment maps. These terms were found in preceding document mining steps and contains the words being nouns, verbs, and adjectives. Note that some of these terms are even compound nouns like ”phone call“ or negated verbs like ”not to send“. The resulting overview visualization can be seen in Figure 5.

On an abstract level there is a clear grouping and ordering of the sentiment maps visible. Terms with only negative occurrences (reddish images) are located in the upper left while positive terms (greenish sentiment maps) are located in the lower right. The first diagonal consists of terms being either mentioned negatively and positively equally often (upper right) and terms with a geospatial, diverse distribution (lower left). The SOM analysis enables the analyst to get a fast overview over terms being mentioned always positive or negative.

The strongly highlighted, white node of Figure 5 in the lower left contains eleven terms showing a very diverse geospatial distribution. As this is the node being highlighted most we will investigate this node in the following paragraphs. Detailed analysis via drill-down techniques are possible in our system and reveal the geospatial distribution for each single term. The visualization of all eleven contained terms is depicted in Figure 6.

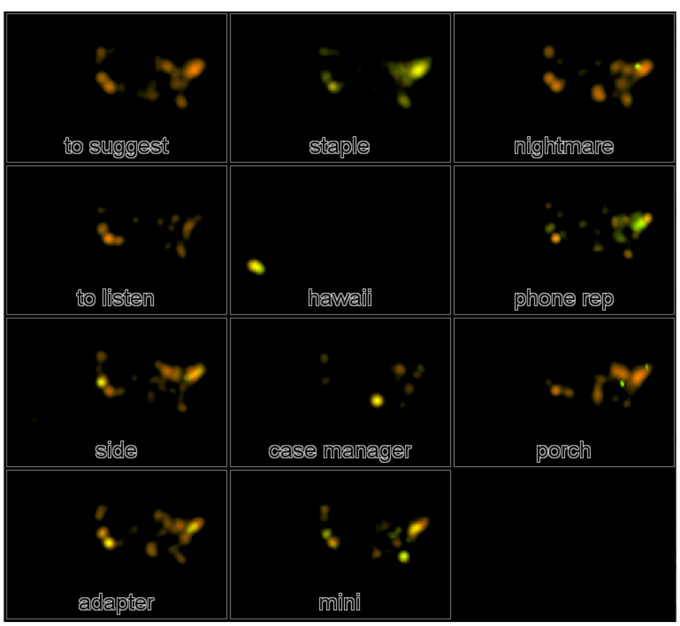


Figure 6. Visual representation of all sentiment maps (in reading order: to suggest, staple, nightmare, to listen, hawaii, phone rep, side, case manager, porch, adapter, mini) contained in the lower left node of Figure 5. [1]

Inspecting the sentiment maps more in detail reveals that this node mainly contains sentiment maps with sparse and diverse geospatial distributions.

The most obvious sentiment map contained in this SOM node is the term ”hawaii“. This term is occurring mostly positively and collocated with the geospatial position of the Hawaiian islands. Inspecting the customer comments in detail, we found that customers liked the free shipping possibilities to Hawaii, which seems not to be taken for granted. Service managers can learn from this information that (Hawaiian) customers do care about the shipping procedure and that free shipping might be an advantage over competitors.

Also, the term ”case manager“ (third row, second column in Figure 6) shows an interesting pattern. Although mostly mentioned negatively because of language issues – the customer support was hard to understand because of foreign accents – there are many positive occurrences in Houston, Texas where customers liked the support regarding their printers. Service managers should now investigate further what the characteristics about the problems in Houston were.

Two further interesting sentiment maps are the ones of ”nightmare“ and ”porch“. Investigating the underlying reviews shows that the preceding sentiment analysis did not work correctly as all the reviews were purely negative. This is not a drawback of the method per se, but exemplifies the uncertainty of any sentiment analysis and the sensitivity of our method to the input data. The comments regarding the term ”porch“ were mentioning that the parcel was left unattended on the porch. The term ”nightmare“ was used in cases where the process of ordering and returning products did not go smoothly.

Temporal Analysis and Findings. We performed a temporal analysis of the sentiment distribution by partitioning the data set according to the time dimension. We used the quarters of a year in order to group the reviews into meaningful units. The analysis is initiated by computing the overall sentiment of the customer feedback messages. Then, we use the resulting sentiment score per message. Figure 7 shows the results. The time dimension is denoted on the x-axis and the sentiment scores on the y-axis. Unfortunately, the differences of the sentiment distributions are not significant as all boxes do overlap each other. Consequently, the overall sentiment distribution does not significantly vary over time when analyzed without any geospatial reference.

We further statistically analyzed and computed the sentiment distribution over time per term. We consequently selected all significant terms contained in the data set and inspected the occurrences. For each of them we assigned the sentiment. The sentiment scores were grouped by the quarter of the year they were received. Then, for each of these groups, we computed the average sentiment. In our case, this lead to the quarters Q3/2007 to Q1/2010 with the related average sentiment scores. As a last step, we compute the variance of these average sentiment scores resulting in a number, that describes, how much the sentiment score changed over time. Sorting this list of terms and sentiment changes over time leads to the most interesting terms, which can be inspected manually.

The simple temporal, statistical analysis described above, helped us in assessing which of the terms show interesting

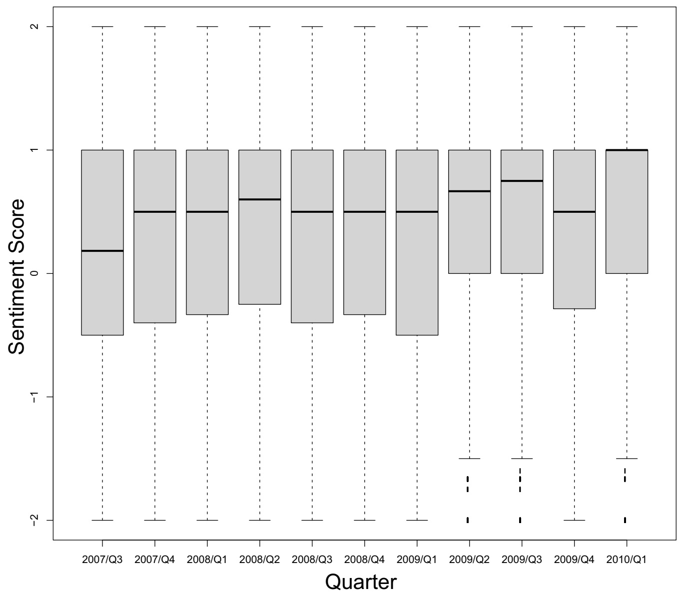


Figure 7. Statistical analysis of the sentiment scores distribution over the different quarters. No significant difference between the distributions is observable.

behavior over time. But these statistical approaches do not reveal interesting findings per se as the geospatial reference is not regarded at all. As described in the technique section above, we partitioned the data set into meaningful time units – in our case into quarters of a year – and computed for these time units sentiment maps. In the following paragraphs, we will show interesting terms found by computing the variance in sentiment scores over the time quarters.

We investigated the top candidates found by the method above and present the corresponding sentiment maps. The first one we inspected is the term *camera*, shown in Figure 8, as it has one the highest variance scores. Comparing the different quarters of the year, it is obvious that only in the first quarter the term *camera* is mentioned positively in California. Interestingly, the second and forth quarters seem quite similar in the sentiment distribution. With this information, the CRM expert can inspect the corresponding feedback messages and plan special actions, such as special offers or sending apology messages.

The second term we present here, is the term *salesman* in Figure 9. Again, this term seems also being very sensitive to the time dimension and is not uniformly distributed. Furthermore, the term salesman seems to be more frequent in the first and last quarter compared to the other quarters. The reason might be that there is more need for guidance in the time around Thanksgiving and Christmas because of presents and salesman are important then.

In order to analyze the temporal development of the sentiment of all terms, we applied the Self-Organizing Map layout to the sentiment maps of one year and place them as small

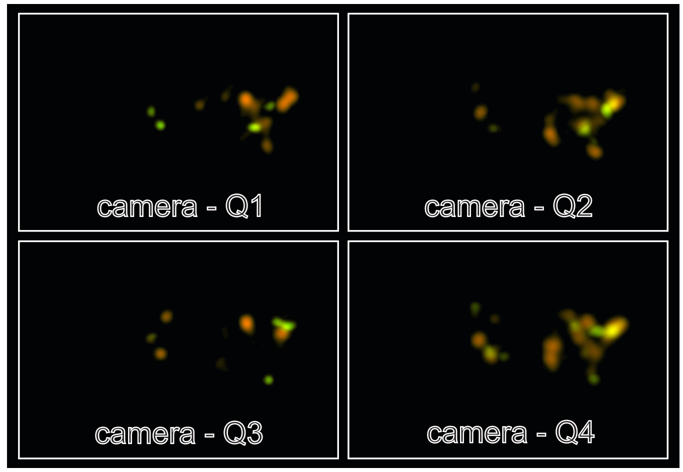


Figure 8. The term *camera* shows one of the highest sentiment variation over time and reveals interesting patterns.

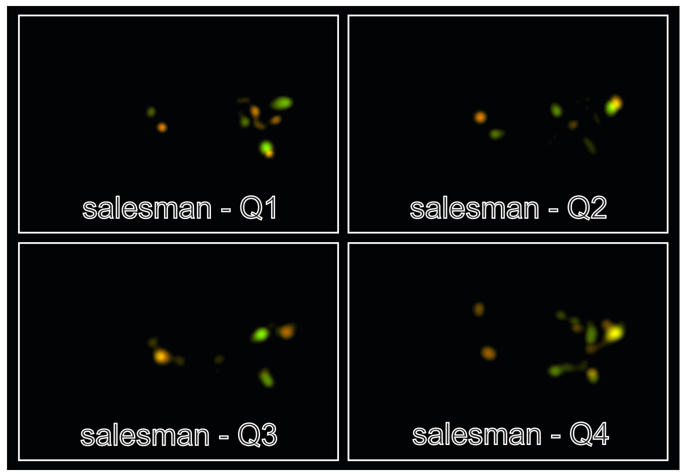


Figure 9. The term *salesman* shows one of the highest sentiment variation over time and reveals interesting patterns.

multiples side-by-side. By comparing the positions of the same term over the different years, a shift in sentiment becomes obvious. We exemplified one of these changes over time in Figure 10. We highlight the SOM node of the same term by Brushing & Linking and investigated in this example the term *customer service*. The term is mentioned quite ambiguous in the first year, but has a quite good overall sentiment distribution in the second year. Unfortunately, it is mentioned worse every year than, with the worst opinion in the last year. This shift of sentiment is visible by the movement of the term from the right side (mostly positive opinions) to the left side (mostly negative comments).

V. Conclusion

We presented our approach to visually compare and inspect large sets of textual customer feedback with respect to sentiment expressed regarding key concepts, and geographic distribution. For each concept, a sentiment map was rendered and

(a) 2007 (b) 2008

(c) 2009 (d) 2010

Figure 10. Comparison of the SOM layouts of the sentiment maps of different years. We highlighted the SOM node containing the term *customer service* by a red rectangle. The sentiment development over time is very good visible, as 2008 is the year with the best opinion about the customer service.

all maps were visually clustered and aggregated by the SOM approach. Interaction methods allow navigating the overview visualizations and drilling down for detailed inspection. Further temporal analyses of the customer feedback revealed seasonal influences on the customer opinion. Application findings presented indicate that key concepts and their sentiment scores being highly dependent on geographic position. Such findings can be very helpful in analyzing service levels across locations, products, customers, and similar applications in CRM. Our analysis system can be easily used on top of existing CRM systems, as we only need the text and the geographic position of the reviews.

We have several ideas to extend our work in future for improved analysis. One possibility improving the visual representation is the integration of semantic zoom approaches. Semantic zoom can allow merging neighboring SOM nodes to reduce the level of detail. Additionally, semantic zoom can be applied to the shown terms by using an ontology. The terms will be grouped by the ontology and only the common parent of a set of related concepts will be visualized. The ontology also leads to another extension possibility we are going to integrate in future. We plan to show the hierarchic relationships between terms directly on the SOM representation. Last but not least, we want to consider more detailed map visualizations concerning production facilities and income distributions among different cities correlating geospatial dependent properties with the text features.

ACKNOWLEDGMENT

This work has been supported by the Consensus project and has been partly funded by the European Commission's 7th Framework Programme through theme ICT-2013.5.4 ICT for Governance and Policy Modelling under contract no.611688.

We are thankful to Ming Hao of Hewlett-Packard Labs for fruitful collaboration on the analysis of geo-referenced customer review report data. We also thank the Helsinki University of Technology for providing the SOMPAK implementation which we used in this study.

REFERENCES

[1] H. Janetzko, D. Jäckle, and T. Schreck, “Comparative visual analysis of large customer feedback based on self-organizing sentiment maps,” Proc. International Conference on Advances in Information Mining and Management, 2013, pp. 12–17.

[2] T. Kohonen, Self-Organizing Maps. Springer-Verlag, 2001.

[3] J. Vesanto, “SOM-based data visualization methods,” Intelligent Data Analysis, vol. 3, no. 2, 1999, pp. 111–126.

[4] G. J. Deboeck and T. K. Kohonen, Eds., Visual Explorations in Finance: with Self-Organizing Maps. Springer, 1998.

[5] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma, “Visual cluster exploration of web clickstream data,” in IEEE VAST, 2012, pp. 3–12.

[6] T. Schreck, J. Bernard, T. von Landesberger, and J. Kohlhammer, “Visual cluster analysis of trajectory data with interactive kohonen maps,” Information Visualization, vol. 8, no. 1, 2009, pp. 14–29, accessed June 13, 2014. [Online]. Available: http://dx.doi.org/10.1057/ivs.2008.29

[7] J. Bernard, J. Brase, D. Fellner, O. Koepler, J. Kohlhammer, T. Ruppert, T. Schreck, and I. Sens, “A visual digital library approach for time-oriented scientific primary data,” Springer International Journal of Digital Libraries, ECDL 2010 Special Issue, 2011, pp. 111–123.

[8] K. U. Barthel, “Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics,” Image Analysis for Multimedia Interactive Services, International Workshop on, vol. 0, 2008, pp. 227–230.

[9] N. Andrienko and G. Andrienko, Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach. Springer-Verlag, 2005.

[10] A. M. MacEachren, How Maps Work - Representation, Visualization, and Design. Guilford Press, 2004.

[11] F. Bao, V. Lobo, and M. Painho, “The self-organizing map, the geo-som, and relevant variants for geosciences,” Computers & Geosciences, vol. 31, no. 2, 2005, pp. 155 – 163, accessed June 13, 2014, geospatial Research in Europe: AGILE 2003. [Online]. Available: http://www.sciencedirect.com/science/article/B6V7D-4F0851H-2/2/ac6bd99e54c5ff81db9a173604b0d3aa

[12] P. Agarwal and A. Skupin, Eds., Self-Organising Maps: Applications in Geographic Information Science. Wiley, 2008.

[13] D. Guo, J. Chen, A. M. MacEachren, and K. Liao, “A visualization system for space-time and multivariate patterns (VIS-STAMP),” IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 6, 2006, pp. 1461–1474.

[14] G. Andrienko, N. Andrienko, S. Bremm, T. Schreck, T. von Landesberger, P. Bak, and D. A. Keim, “Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns,” Computer Graphics Forum, vol. 29, no. 3, 2010, pp. 913–92.

[15] R. Kincaid, “SignalLens: Focus+context applied to electronic time series,” IEEE Trans. on Visualization and Computer Graphics, vol. 16, 2010, pp. 900–907.

[16] J. Van Wijk and E. Van Selow, “Cluster and calendar based visualization of time series data,” in Proc. IEEE Symposium on Information Visualization. IEEE Computer Society, 1999, pp. 4–.

[17] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, Visualization of Time-Oriented Data. Springer-Verlag New York Inc, 2011.

[18] J. Bernard, J. Brase, D. Fellner, O. Koepler, J. Kohlhammer, T. Ruppert, T. Schreck, and I. Sens, “A visual digital library approach for time-oriented scientific primary data,” Springer International Journal on Digital Libraries, ECDL 2010 Special Issue, 2011, pp. 1–13.

[19] M. Ward, “A taxonomy of glyph placement strategies for multidimensional data visualization,” Information Visualization, vol. 1, no. 3/4, Dec. 2002, pp. 194–210.

[20] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim, “Visual readability analysis: How to make your writings easier to read,” IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 5, May 2012, pp. 662–674.

[21] P. Juola, “Authorship attribution,” Found. Trends Inf. Retr., vol. 1, no. 3, Dec. 2006, pp. 233–334, accessed June 13, 2014. [Online]. Available: http://dx.doi.org/10.1561/1500000005

[22] C. Rohrdantz, M. C. Hao, U. Dayal, L.-E. Haug, and D. A. Keim, “Feature-based visual sentiment analysis of text document streams,” ACM Transactions on Intelligent Systems and Technology, Special Issue on Intelligent Visual Interfaces for Text Analysis, vol. 3, no. 2, 2012, pp. 26:1–26:25.

[23] M. C. Hao, C. Rohrdantz, H. Janetzko, D. A. Keim, U. Dayal, L.-E. Haug, M. Hsu, and F. Stoffel, “Visual sentiment analysis of customer feedback streams using geo-temporal term associations,” Information Visualization, Jun. 2013.

[24] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, “SOM_PAK: The self-organizing map program package,” Helsinki University of Technology, Tech. Rep. A31, 1996.

[25] T. Porter and T. Duff, “Compositing digital images,” in Proceedings of the 11th annual conference on Computer graphics and interactive techniques, ser. SIGGRAPH ’84. New York, NY, USA: ACM, 1984, pp. 253–259, accessed June 13, 2014. [Online]. Available: http://doi.acm.org/10.1145/800031.808606

Towards Establishing an Expert System for Forensic Text Analysis

Michael Spranger and Dirk Labudde
Department of Mathematics, Physics, Informatics
University of Applied Sciences Mittweida
Mittweida, Germany
Email: {*michael.spranger, dirk.labudde*}@hs-mittweida.de

***Abstract*—The analysis of digital media and particularly texts acquired in the context of police securing/seizure is currently a very time-consuming, error-prone and largely manual process. Nevertheless, such analysis are often crucial for finding evidential information in criminal proceedings in general as well as fulfilling any judicial investigation mandate. Therefore, an integrated and knowledge-based computational solution for supporting the analysis and subsequent evaluation process is currently developed by the authors. In this work, we outline the main ideas of this framework and present an approach for categorizing texts with adjustable precision combining rule-based decision formula and machine learning techniques. Furthermore, we introduce a text processing pipeline for deep analysis of forensic texts as well as approaches towards solving domain specific problems like detection and understanding of hidden semantics as well as the automatic assignment of forensic roles.**

Keywords–forensic, ontology, German, text processing, expert system, text analysis, Topic Map, categorization, classification

I. Introduction

The analysis of texts that are subject of legal considerations with the goal of obtaining criminalistic evidence is a branch of general linguistics [1], [2]. Such texts are retrieved by persons involved in the criminal proceedings from a variety of sources, e.g., secured or confiscated storage devices, computers and social networks. Forensic texts, as considered in this work, relate to textual data that may contain evidential information. In contrast to the texts usually considered in scientific work focussing text processing tasks, this kind of texts are neither clearly defined nor thematically unified. Additionally, such texts may vary in quality with respect to their grammar, wording and spelling, which strongly depends on the author's language skills and the target audience. Rather, textual data of different type and origin need to be meaningfully linked to answer a specific criminalistic question reasonably and above all accurately. Furthermore, forensic linguistics cover beside other research topics, utterance and word meaning or authorship analysis and proof [3].

The results of these analyses are used to solve other more complex problems in the criminal investigations, like

- recognition and separation of texts with a case-related criminalistic relevance
- recognition of relations in these texts in order to reveal whole relationship networks and planned activities
- identification and/or tracking of fragmented texts
- identification or tracking of hidden semantics

In the considered context, the term *hidden semantics* is synonymous with one kind of linguistic steganography but not restricted to this. Rather, even the use of slang afflicted language let known text mining algorithms fail. Understanding hidden semantics is one of the hardest tasks during the analysis of forensic texts not only for machines but also for humans. Among other things, for this reason, this kind of deep analysis takes a long time, especially if the amount and heterogeneity of data, the fast changeover of communication forms and communication technologies is taken into account. In order to solve this problem, computer linguistic methods and technologies can be applied. These are originated in the crossover of linguistics and computer sciences [4]. The complexity of the evaluation makes it difficult to develop one single tool covering all fields of application. In order to address this problem, a domain framework is currently under development (see [5] for further discussions).

As a consequence of the analysis of the secured data from a historical case of business crime and the exploration of the special needs of criminologists discussed in Section II, we present in this work a pipeline for categorizing texts with adjustable precision using an approach that is a combination of rule-based decision formula and machine learning techniques. Especially, that leaves the opportunity to the criminologist to decide whether the specificity (precision) or the sensitivity (recall) is more important. Although a high sensitivity may be of greater practical importance. Thus, a high sensitivity is principally necessary to find all incriminating or even exculpatory documents but the results need to be filtered manually since they may be interspersed with irrelevant documents, whereas a high specificity is sometimes more appropriate to get a quick overview about the corpus. Furthermore, we outline a text processing pipeline for deep analysis of forensic texts based on these insights and a rule-based approach for identifying special roles of named entities. Subsequently, we introduce two approaches towards solving the hidden semantics problem. Currently, the text categorization module is evaluated in practice whereas the deep analysis pipeline including the role identification as well as the hidden semantics detector is under implementation.

In the next Section, the peculiarities of the considered kind of texts is shown at a glance. In Section III, a special crime ontology acting as foundation model for an expert system in the field of forensic text analysis is presented. Subsequently, in Section IV a pipeline for analysing forensic texts deeply as well as a first approaches for detecting forensic roles and hidden semantics is outlined before a practicable method for categorizing such texts is introduced and discussed.

II. Assessment of Requirements

This work focusses textual data secured by persons involved in the criminal investigations as part of the evidence process. Hence, for the purposes of this work historical data in a case of business crime is provided by the local prosecutorial. A first manual assessment of these data enables to determine, whether:

- the data material is of considerable heterogeneity related to its structure and domain
- important information may be situated in non-text based data (e.g., photocopies of invoices)
- there are irrelevant texts that may hide relevant information through their abundance (e.g., forms, templates)
- information may have been deliberately obscured in order to protect them from discovery
- some texts can be characterized by strong syntactic weaknesses
- some texts may be fragmented by erasing/reconstruction

These specific characteristics distinguish the examined corpus from other corpora commonly used and evaluated in research.

Further, a survey made by the authors, which was conducted by affiliated criminalists, has revealed that finding and separating relevant documents seized in the database is the most time consuming and difficult part during the evaluation.

III. Development of a Crime Ontology

A. Ontology-based Information Extraction

The term *ontology* is commonly understood as a formal and explicit specification of a common conceptualization. In particular, it defines common classified terms and symbols referred to a syntax and a network of associate relations [6], [7]. Developing ontologies for criminalistic purposes is a prior condition for annotating texts and raise questions in this particular domain. The term *taxonomy* as a subset of ontology is used for the classification of terms (concepts) in ontologies and documents. On the one hand, a criminalistic ontology is characterised by its case-based polymorphic structure and on the other hand by special terms used in criminal proceedings. This aspect has to be taken into account by the definition of any ontology representation model, as we see in Section III-C. Ontologies can be divided into two levels of generality. A *domain ontology* models the knowledge of an almost highly-specialised domain as a part of the real world in an extensive and profound manner. An *upper ontology* describes the common objects applicable to a wide range of domain ontologies. Furthermore, it creates a glossary of basic terms and object descriptions used in various relevant domains [7].

Cowie and Wilks [8] constitute Information Extraction (IE) as a process for selectively structuring and combining data, located, explicitly stated or implied in various texts. A slightly more formal view is given by Russell and Norvig. They understand IE as the acquisition of knowledge by searching occurrences of objects of specific classes and relations between them within natural language text [9].

The process of IE can be supported by ontologies in several ways. The usage as *extraction ontology* is one way to participate in the benefits of ontologies. In this case the IE process itself is guided by using templates generally used by sophisticated techniques of knowledge representation [7], [10]. Presenting the output of the IE process using ontologies is another way supporting this process.

Combining both approaches we obtain an IE system that is supported at most by ontologies. Such systems are called Ontology-based Information Extraction (OBIE)-systems [10].

B. Representation of Knowledge Models

The representation of ontologies can be realized through different models with different levels of expressiveness. Taxonomies and thesauri, which are not mentioned here, can be considered as simple ontologies under the adherence to certain conventions. Instead of this, some more expressive models will be introduced in this section.

The intention of concept maps, as developed by Josef Novak at the Cornell University [11], is to represent relationships between concepts. According to this, a concept map is an abstract description of certain ideas or of a specific knowledge domain. They visualize semantic units (prepositions) for a certain domain, while semantic units consist of two terms (concepts) connected through a named relation. Labelling a relation provides a higher degree of understanding through additional semantic information. It is explicitly not forbidden to create cross relations between multiple concepts [7], [11].

Topic map is the most expressive model and well defined because of its ISO standardisation. There is a wide variety of implementations, e.g., XML Topic Maps (XTM), transposing the basic concepts of this standard although they ignore or modify single aspects defined by the ISO standard.

The standard *ISO/IEC 13250* describes the usage of topic maps in the areas of information exchange, organization and representation with the aid of topics. Basically, structural information provided by topic maps allow to describe relations between topics, related to abstract things, and to attach addressable information objects to a single topic (occurrences). The nature of all constituent parts can be described more in detail by using properties (facets). Another significant point is that the information objects used in a topic map can be assigned to a scope as described in more detail in Section III-C. It is important to know that several topic maps can provide structural information referring to the same resource. In this way, the architecture enables the combination of topic maps and the coupling of information from different areas. Because of their extrinsic character topic maps can be seen as an extension or overlay of information objects. In summary it can be stated that topic maps enable versatile and simultaneous views at information objects, whose structural nature is principally unrestricted. Hence, it is possible to use an object-oriented, hierarchical, sorted or unsorted approach or each combination

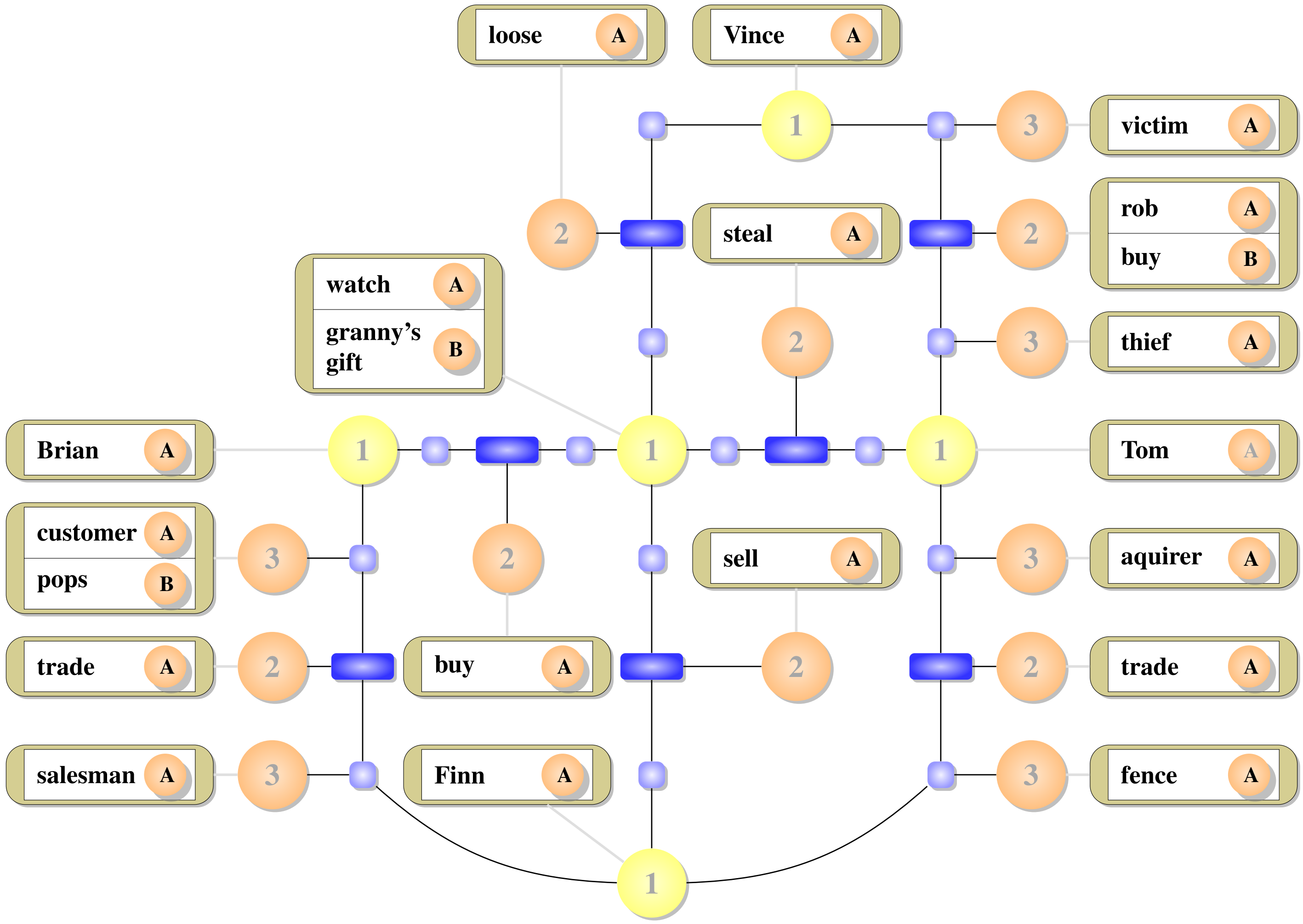


Figure 1: Extract of an ontology used for the description of property crimes. It demonstrates a typical interaction of the different topic map elements, whereas familiar relations are not included here.

of these. Additionally, it is possible to overlay an unrestricted count of topic maps on a given set of information resources [12].

C. Crime Ontology Model

In this project we use a modified variant of the topic map standard to model an ontology, where the created model is based on the contents and thoughts of the ISO standard without claiming a full implementation of all parts. In general, major semantic elements can be considered to be present in the model while most syntactic elements have been replaced with elements as required by a model driven software development. Especially, the use of *scopes* within topic maps is a significant advantage for modelling multilingualism and improves the determination of meanings. In the field of crime sciences and forensic linguistics multilingualism is not only restricted to native and foreign languages, moreover it is possible to integrate slang afflicted language groups, dialects and different verbal skills. Furthermore, *scopes* offer one possibility to solve the hidden semantics problem, we considered in Section I, by annotating one or more different meanings directly to the particular topic. The topic map elements used in the model considered here are described in Table I.

Figure 1 demonstrates an application of the topic map derivative as developed under this work for modelling a criminalistic ontology – a simple case of uncovering a ring dealing with stolen goods. The shown extract does not cover all elements of the topic map model implemented.

The core objects in the example network are highlighted by the number *1* – the persons Vince, Tom, Finn and Brian, as well as the item watch. Associations specified through descriptive topics between these objects are highlighted with the number *2*. A specified role, taken by an object within an association, is highlighted with the number *3*.

Taking a closer look at the example shown in Figure 1 leads to the suggestion that the course of creating this network could have been happened the following way: Brian is searching for a watch because his old one is broken. He asks in different stores for a model fitting his needs till he finds a salesman (Finn) who offers him that he might get one in his next delivery. A few days later Finn calls Brian that he got a watch for him, Brian does not hesitate and buys it. After a closer examination at home he comes over a nearly faded

TABLE I: Elements of the forensic Topic Map model

Element	Description
Subject (Topic)	an abstract or concrete entity in the domain to be analysed
Instance (Topic)	the concrete manifestation of a subject (*red circle*)
Descriptor (Topic)	typifies any other syntactical elements (*orange circle*); i.e., adds further details related
Association	a relation between two topics, usually subject and instance (*light blue rhomboid*)
Association Role	specifies the roles of the topics in an association (*blue square*)
Occurrence	corresponds to the crete manifestation of a topic in a resource, usually related to an Instance.
Topic Name	is the name representation of topics (*green rounded rectangle*)
Name Item	denotes the name of a specific topic, associated to a Scope (*white rectangle within the topic name*)
Facet	names a class of attributes of a topic and can include several Facet Values
Facet Value	a particular attribute as distinct value; can be a topic or another Facet
Scope	defines semantic layers; e.g., causing system to focus by filtering particular syntactical elements

inscription on the back of the watch and shows it his friend, a policeman. Some days earlier the policeman was called by Vince, a person who lost his heirloom at the beach, which has an inscription just like this watch. They went back to the store together where Finn was spotted by the policeman, known to him from smaller complaints by different customers. After some consideration time the police confiscated Finns computer. Within the analysis of the confiscated material an instant messaging protocol reveals the following snippet:
Tom: "I bought granny's gift, which pops demanded."
Finn: "Alright, bring it over."
Where Tom is also known to the police with no familiar relations to Finn. Some further background work reveals the full potential of their relation and completes the network. Reconsidering all the facts Finn can be marked as a fence who sells stolen goods acquired by Tom. He kept looking for a watch described by Brian and finally found a model easy to steal, Vince's watch. Lucky coincidence in this posed example for demonstrating the cooperation of the different elements of the ontology model to uncover a fence network.

IV. Approaches in Forensic Text Analysis

In this section, several strategies for handling forensic texts respecting the insights from the needs assessment (Section II) are introduced. Since the most aspects of this work are currently under implementation no final results will be presented yet. Thus, these aspects are only outlined subsequently.

A. Pipeline for Deep Analysis

The deep analysis of forensic texts has to respect their characteristics described in the previous section. It includes particularly tasks in Information/Event Extraction to instantiate a criminological ontology as the central element in the solution developed under this work. In particular, the work of Wimalasuriya and Dou [10], Embley [13] and Maedche [14], shows that the use of ontologies is suitable for assisting the extraction of semantic units as well as their visualization and structures such processes very well. We have divided the whole process in three sub-processes:

1) creation of both the criminological ontology and the analysis corpus
2) basic textual processing and detection of secondary contexts
3) instantiation of the ontology and iteratively refinement

In order to define the extraction tasks as well as to introduce case-based knowledge the first of all is the creation of the criminological ontology in its specialized form as Topic Map, which we have developed in an earlier work [5]. This step may be supported by using existing ontologies created in similar previous cases. Subsequently, the analysis corpus needs to be created, especially for separating the textual data from other files and extracting the raw texts from the documents also including optical character recognition in cases of digital images like photocopies. This data is stored in a database together with extracted meta-data and added to an index for quick access. In the second step some state-of-the-art textual processing steps like Part-of-Speech-tagging, language recognition and some special operations for structured texts may be performed. Especially, we detect event-narrative documents. This task has been introduced by Huang and Riloff [15] for exploring secondary contexts. They define these as sentences that are not explicitly part of the main event description. Nevertheless, these secondary contexts could yield information related to the event of interest that could provide important evidence or lead to the booty, further victims or accomplices. The final step within the main process is constituted by the actual extraction process. Here, the actual event sentences that are suitable to instantiate at least one part of the ontology are recognized and, if needed, extracted together with the information from secondary contexts. Then, we try to refine the instantiated model iteratively by identifying forensic roles as described in IV-B. Figure 2 illustrates the whole process schematically.

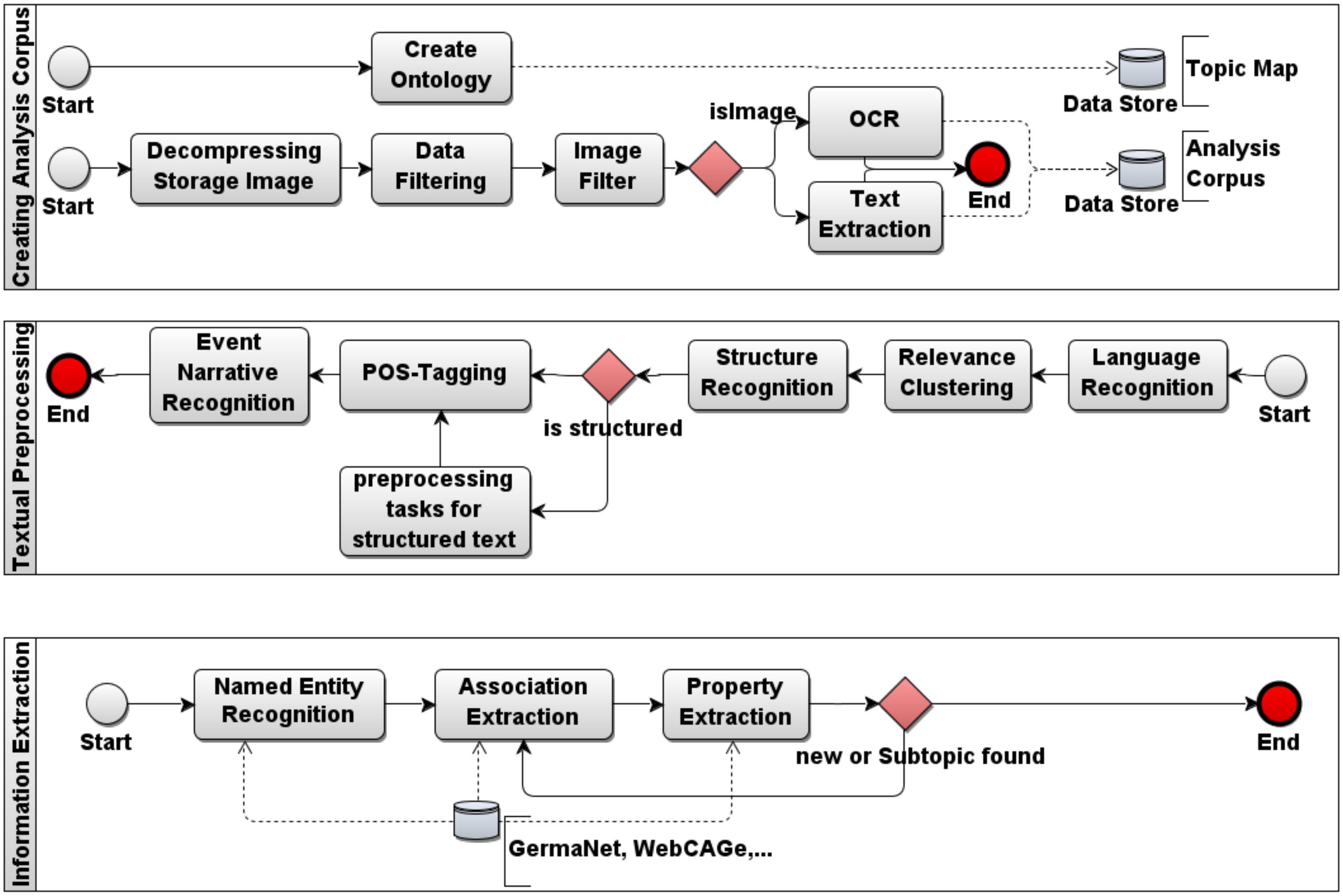


Figure 2: The tool-pipeline for deep analysis. We have divided the whole process in three sub-processes: 1) creating analysis corpus 2) textual preprocessing 3) information extraction

B. Identification of Forensic Roles

The recognition of named entities is a well-researched part of Text Mining and a regular task in every Information/Event Extraction solution as well as in our pipeline mentioned in Section IV-A. The general task is to identify all instances $i \in I$ of each concept $c \in C$ taking into account their hypernymy and hyponymy relationships. This task can be solved practically by using Gazeteer-based solutions via supervised learning methods [16], [17] up to the usage of semi-/unsupervised learning approaches [18]. However, no existing solution we applied has been proven itself to be able to assign forensic roles. The assignment of such a role often depends on more than one document as well as on the contribution of case-based knowledge by the criminalist. Therefore, our framework is based on an ontology acting as an extraction and visualization template that is able to provide such knowledge. The ontology model we used is based on the Topic Map standard. In our previous work [5], we stated that each topic can contain a set of facets. These facets are used beside others to model rules that an inference machine can use to reason the appropriate role of an entity within a post-process. In this way, the level of detail within the computational recognition of entities is able to be increased. Figure 3 shows a detail of a fictional forensic Topic Map that could have been created by a criminalist. Here, a accomplice is described as a person that satisfies one or two of the following rules:

- the person has common interest in the deed exactly when he has instantiated an association possess with the instance of a topic acting in the role of booty
- the person has shared worked exactly when their related instance in the Topic Map has an instantiated association drive to an instance of the topic car acting in the role of a means of escape

The number of rules that have to be satisfied depends on rule weights, which act as indicators for rule importance. The concrete instance defines the same facets with binary values depending on the matching behaviour of each rule.

C. Towards Solving the Hidden Semantics Problem

As mentioned in Section I the hidden semantics problem is one of the hardest tasks during the analysis of forensic texts even for criminalists or linguistic experts with years of experience. Thus, this problem can only be solved by consideration of the whole context and the knowledge of experts. A system that should be able to detect or even solve this problem automatically needs to process the overall IE-tasks before. Since knowledge extracted automatically as well

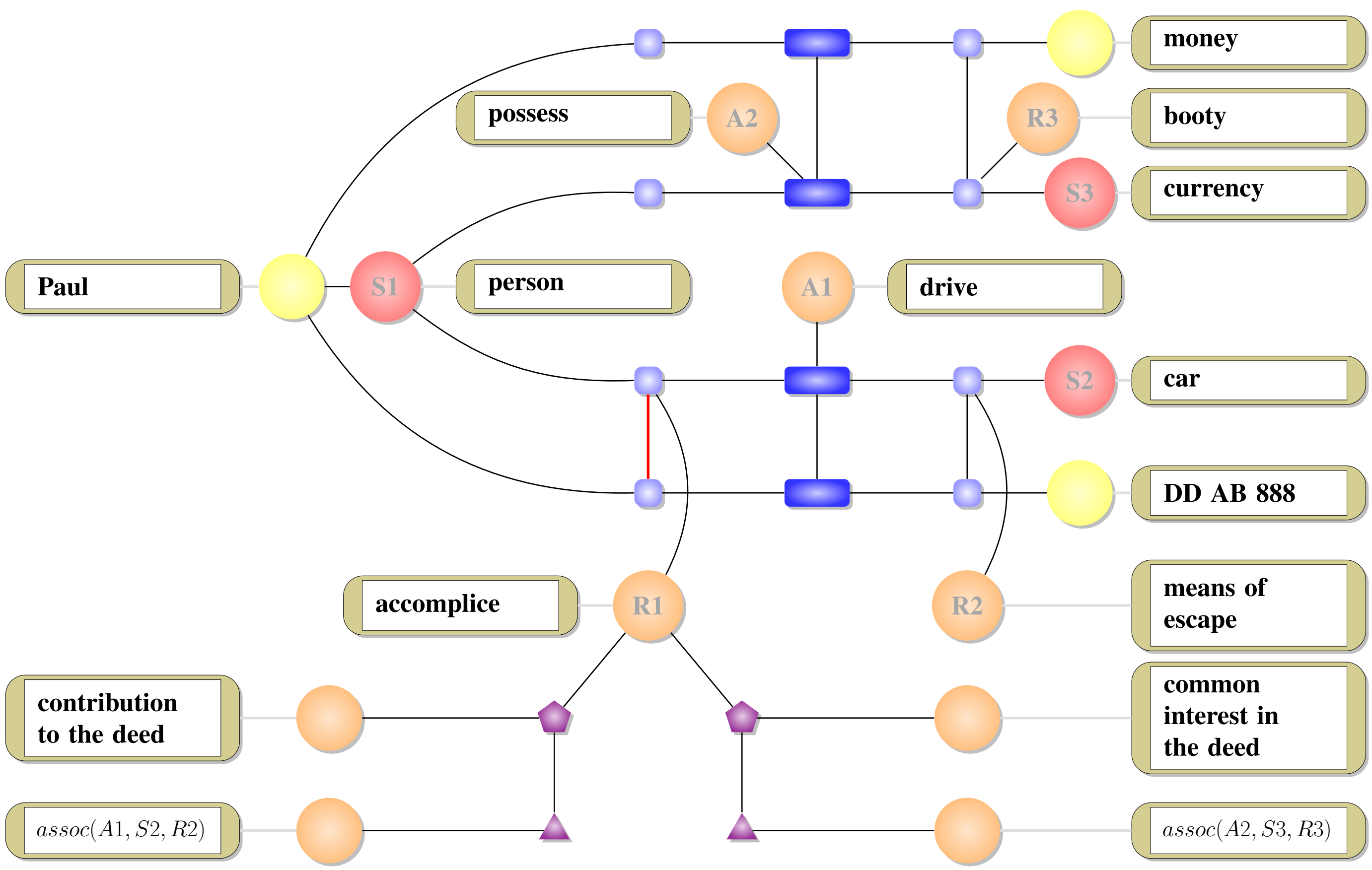


Figure 3: Gradually refining of named entities. The entity *Paul* as instance (yellow circle) of the abstract topic (red circle) *person* can be gradually assigned to their concrete manifestation *accomplice*, which is a subtopic by iterative comparison of its facets lodged as rules.

as introduced by experts is represented by a criminalistic Topic Map (see Section III-C), hidden semantics might be detected by considering its special features. Maicher has introduced an approach for merging Topics with the same meaning modelled by different authors in an distributed world [19]. This leads to a similar approach for the problem discussed here. Thus, each instance a system may find is clearly defined by the position of the related topic within the taxonomy, its facets and the set of instantiated associations where it plays a highly specific role. We assume this semantic context will remain approximately constant if the text is transposed towards a steganographic code, because only the wording changes (see Figure 4a).

More formal, let each Instance $i \in I$ be well defined by a tupel $\{T, F_T, R_A, A_T\}$, where T is the related Topic-hierarchy, F_T is a set of Facets of each of this Topics that discriminates the instance from other similar ones, R_A is a set of Roles that it plays relating to a set of Associations and finally A_T is a set of Associations of each Topic. This tupel constitutes the context $C(i)$ of a specific Instance. Subsequently, each context has to be compared with the context of other Topics using a distance function $dist$ to find out the degree of similarity. The definition of a threshold ϵ supports the decision, whether two topics are possibly the same or not [see equations (1) and (2)].

$$\Delta_{min}(C(i)) = min_{j \in I \setminus i}\{dist(C(i), C(j))\} \qquad (1)$$

$$SYN(C(i), C(j) = \begin{cases} 1, & C(j)\ has\ \Delta_{min}(C(i)) < \epsilon \\ 0, & else \end{cases} \qquad (2)$$

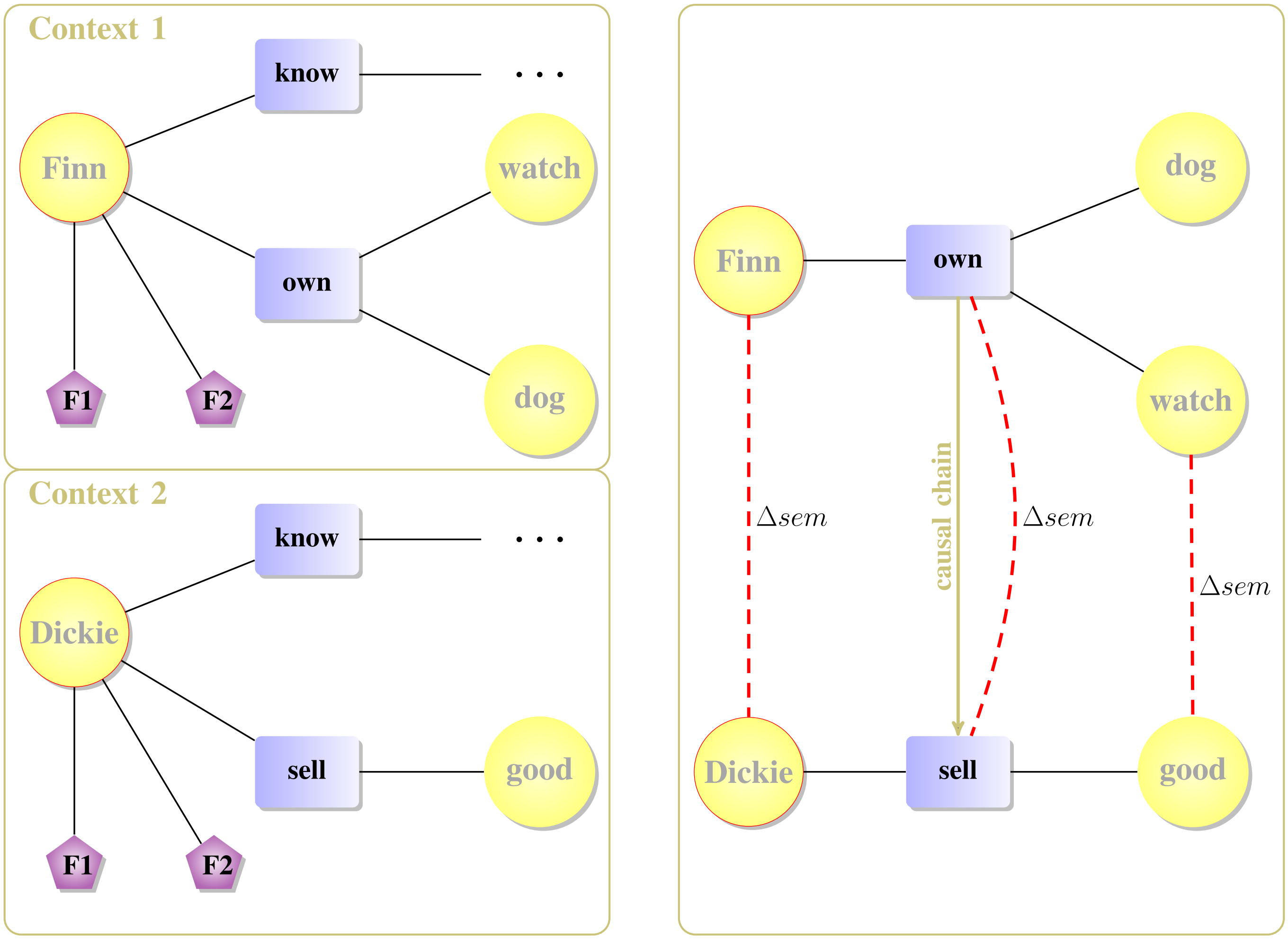
In order to determine the distance between contexts the semantics in the ontology need to be encoded in a numeric format. For Topics the method of Wang et al. [20] can be adapted, whereby the farther away from one Topic to another, the less similarity is determined by the constant k [see equation (3)]. This constant needs to be determined empirically.

$$S_T(t) = \begin{cases} 1, & t = T \\ max\{k * S_T(t') \mid t' \in children(t)\}, & else \end{cases} \qquad (3)$$

Another approach is more Association-centred. We consider alignments of all Associations within the same causal chain and calculate an edit distance. This distance measure is related to distances in the ontology-graph (see Figure 4b). Formally, let A be the set of associations and K the set of causal chains that may be derived from A. A causal chain is constituted by all associations $\{a_1 ... a_n\}$, whereby $a_1 \rightarrow a_2 \rightarrow ... \rightarrow a_n$. Further, let A_T be the set of Associations related to an specific Topic. The causal chains in that we are interested in can be described as

$$K_{relevant} = \{k \in K \mid \exists a, b \in k \wedge a \in A_{T1} \wedge b \in A_{T2}\} \qquad (4)$$

Let S be the set of sentences that can be built using any association in one k. Thus, we can calculate a score for each



(a) Topic-centred approach - Assuming the two instances *Finn* and *Dickie* referring to the same entity. Each of these defines a context that contains the facets, associations and roles derived from the topic they are associated with. The level of similarity between the two contexts also indicates the similarity of the instances.

(b) Association-centred approach - Assuming the two instances *Finn* and *Dickie* are the same. The similarity of the two instances can be determined by pairwise alignment of all possible sentences and calculating a semantic distance. Since the associations *own* and *sell* are situated in one causal chain the probability for synonymity of the instances will increase.

Figure 4: Detection of Hidden Semantics

alignment $\{(a, b) \mid a, b \in S\}$. The higher this score the higher the probability that the Topics involved have the same meaning.

D. Categorization of Forensic Texts

As discussed in Section II, filtering and categorization is the most important task in evaluation of forensic texts and a regular Information Retrieval task. Categorization as a specialization of classification aims to place a document in one small set of categories using machine learning techniques. More formal, given a set of documents $D = \{d_1, ..., d_m\}$ and further a set of categories $C = \{c_1, ..., c_n\}$ the task can be described as an surjective mapping $f : C \to D$. Ikonomakis et al. [21] have given an overview about supervised machine learning methods for solving this problem. However, they observed that the performance is significantly depending on a corpus of high quality and sufficient size. Riloff and Lehnert [22] introduced an approach for high-precision text classification. The augmented relevancy signature algorithm they introduced reached up to 100% precision with over 60% recall on the MUC-4 corpus [23]. Nevertheless, in the focussed domain these results are not always sufficient, especially since they do not relate to the properties of forensic texts. It has to be emphasized, that each false-negative (a not identified, case-relevant document) could provide crucial evidences. This highlights the necessity for a method that yields at best 100% in sensitivity with justifiable precision. Beebe and Clark [24] have introduced an approach to handle the information overload resulting from the sensitivity-precision trade-off problem. They considered a similar problem and suggest to cluster the results thematically. However, designing and training a suitable classifier is a challenging problem. Due to the fact that the knowledge of the criminalist (general and case-based) is available related to a concrete judicial investigation order, rules can improve the performance in some cases. Since the categories are modelled as a taxonomy tree we can extend this model so that we are able to assign a set of rules (e.g., regular

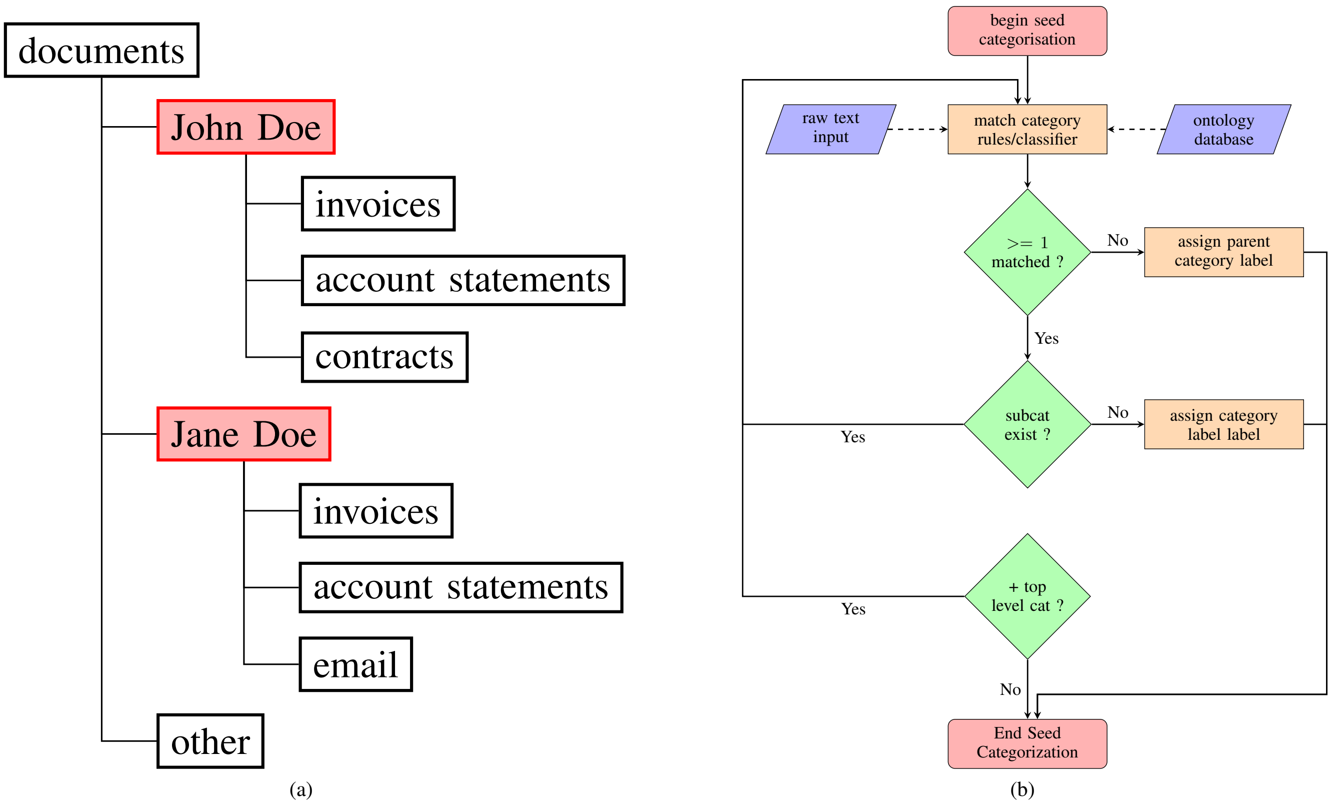


Figure 5: *Acquisition of seed documents:* The raw text under consideration is checked against a set of category rules recursively. Starting at a top-level category, at least one category rule/classifier has to match until the match of each subcategory, drawn from recursion, has failed. In this way, only the label of the most specific category starting at each existing top-level category is assigned.

expressions applied on the documents body) to each category. These rules are combined by disjunction within the categories itself and by conjunction between different categories in cases of one continuous chain of parent-child relationships (Figure 5a). Each of these rules has to define the target that it should applied on (e.g., file name or content), a rule type that helps to select the corresponding rule solver and the rule itself. In this way, we are able to select a certain number of seeds that ensure high precision, which is required to start an appropriate bootstrapping machine learning algorithm to classify the remaining documents (Figure 6). The whole selection process of seed documents is shown in Figure 5b. Notice, the performance of the machine learning algorithm used can be influenced by rephrasing the corresponding rules, since the performance of a bootstrapping algorithm significantly depends on the seed elements chosen, more precise their representativeness. Thus, strictly formulated rules may result in high precision but low sensitivity, whereas applying more weak rules will increase the sensitivity.

First measures of performance using probability-based classifiers, like Naive Bayes, as well as similarity-based classifiers, like k-NN or TF-IDF shows that the performance reaches up to 100% precision with 93.58% sensitivity applied on the corpus provided by the prosecutorial as mentioned in Section II. The results are depending on the employed algorithm and the concrete category and could be a consequence of classifier over-fitting caused by the underlying homogeneous corpus. We have observed that in the in the corpus we used the documents are characterized by remarkable similarity. Therefore, a more appropriate corpus is created currently. For lack of an additional real-life corpus we cross-checked our results using a subset of the 20-Newsgroups-Corpus [25] consisting of the categories *med* and *space*. Depending on the chosen start-rules we achieved sensitivity between 87.6% and 92.4% with precision between 52.3% and 100% (F1 66.79% - 93.39%). This result confirms the strong dependence of the rules used. One of the biggest advantages of this combined approach lays in the adjustable precision depending on an intelligent combination of rules and machine learning algorithms.

V. Conclusion

In this work, we have outlined some kernel processes for information extraction in the environment of the criminal proceedings. These processes are suitable to deal with very heterogeneous data concerning their domain as well as their quality. In the task of deep exploration of the raw data we put great emphasis on the discovery of all relevant information using secondary contexts to avoid misunderstandings and lacks in the evidence. In the identification of forensic roles we have described a new approach in refining ontology instances by deriving and applying semantic roles logic-based. A corresponding module using the logic programming

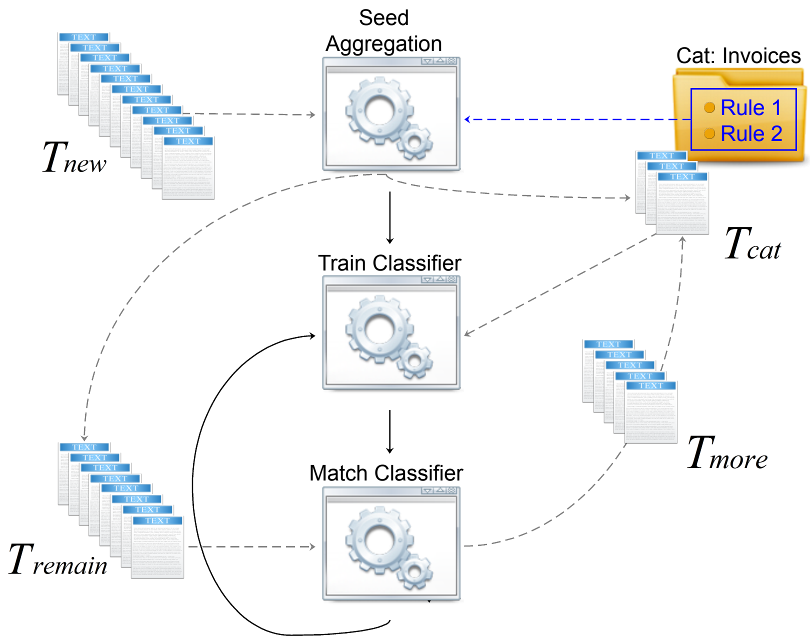


Figure 6: Bootstrapping algorithm for classifying forensic texts. From the texts T_{new} a set of seed documents for each category is acquired using the rules annotated in the taxonomy. This set T_{cat} is used to train one initial weak binary classifier per category. Subsequently, this classifier is used to classify the remaining texts T_{remain} and store the new labelled documents T_{more} to T_{cat}. Finally, the classifier is going to be improved iteratively using T_{cat} until no document is left or no further improvement is possible.

language *Prolog* is currently under development. Furthermore, we introduced two approaches towards solving the hidden semantic problem. Both are based on the calculation of a semantic distance measure using the forensic topic map model we presented at the very beginning. In the task of classification of forensic texts we have to respect that each misclassified file could lead to a lack of evidence. Therefore, it must be ensured that at best no type II errors occur during the categorization. At the same time the taxonomy definition has to remain flexible. Because of a lack of training data supervised learning is not applicable. Therefore, a bootstrapping approach is chosen, combined with a rule-based search for seed files we have earned very good preliminary results up to 100% precision with 93.58% (F1 = 96.68%) sensitivity in selected domains. However, this unexpected result could be due to an over-fitting to the used corpus. For this reason we currently creating a new extended corpus with the support of the local prosecutorial.

ACKNOWLEDGMENT

The authors would like to thank the members of the criminal investigation department and the prosecutorial Chemnitz (Germany). We acknowledge funding by "Europäischer Sozialfonds" (ESF), the Free State of Saxony and the University of Applied Sciences Mittweida.

REFERENCES

[1] M. Spranger and D. Labudde, "Semantic tools for forensics: Approaches in forensic text analysis," in Proc. 3rd. International Conference on Advances in Information Management and Mining (IMMM), IARIA. ThinkMind Library, 2013, pp. 97–100.

[2] H. Kniffka, Working in Language and Law. A German perspective. Palgrave, 2007.

[3] E. Fobbe, Forensische Linguistik - Eine Einführung. Narr Frankcke Attempto Verlag, 2011.

[4] K.-U. Carstensen, C. Ebert, C. Ebert, S. Jekat, R. Klabunde, and H. Langer, Computerlinguistik und Sprachtechnologie - Eine Einführung, 3rd ed. Spektrum Akademischer Verlag, 2010.

[5] M. Spranger, S. Schildbach, F. Heinke, S. Grunert, and D. Labudde, "Semantic tools for forensics: A highly adaptable framework," in Proc. 2nd. International Conference on Advances in Information Management and Mining (IMMM), IARIA. ThinkMind Library, 2012, pp. 27–31.

[6] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in Formal Ontology in Conceptual Analysis and Knowledge Representation, N. Guarino and R. Poli, Eds. Kluwer Academic Publishers, 1993.

[7] A. Dengel, Ed., Semantische Technologien, 1st ed. Spektrum Akademischer Verlag, 2012.

[8] J. Cowie and Y. Wilks, "Information extraction," in Handbook of Natural Language Processing., H. M. R. Dale and H. Somers, Eds. New York: Marcel Dekker, 2000.

[9] S. Russell and P. Norvig, Künstliche Intelligenz: Ein moderner Ansatz, 3rd ed. Paearson Deutschland, 2012.

[10] D. C. Wimalasuriya and D. Dou, “Ontology-based information extraction: An introduction and a survey of current approaches,” Journal of Information Science, vol. 36, no. 3, 2010, pp. 306–323.

[11] J. D. Novak and D. B. Gowin, Learning how to learn. Cambridge University Press, 1984.

[12] ISO/IEC, Topic Maps, Information Technology, Document Description and Processing Languages, ISO/IEC Std. ISO/IEC 13 250, Rev. Second Edition, 2002.

[13] D. W. Embley, “Toward semantic understanding: an approach based on information extraction ontologies,” in Proceedings of the 15th Australasian database conference - Volume 27, ser. ADC ’04. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 3–12.

[14] A. Maedche, G. Neumann, and S. Staab, “Bootstrapping an ontology-based information extraction system,” Studies In Fuzziness And Soft Computing, vol. 111, 2003, pp. 345–362.

[15] R. Huang and E. Riloff, “Peeling back the layers: detecting event role fillers in secondary contexts,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, ser. HLT ’11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1137–1147.

[16] A. Mikheev, M. Moens, and C. Grover, “Named entity recognition without gazetteers,” in Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, ser. EACL ’99. Stroudsburg, PA, USA: Association for Computational Linguistics, 1999, pp. 1–8.

[17] G. Zhou and J. Su, “Named entity recognition using an hmm-based chunk tagger,” in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ser. ACL ’02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 473–480.

[18] Z. Kozareva, “Bootstrapping named entity recognition with automatically generated gazetteer lists,” in Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, ser. EACL ’06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 15–21.

[19] L. Maicher, “The impact of semantic handshakes,” in Proceedings of the 2nd international conference on Topic maps research and applications, ser. TMRA’06. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 140–151.

[20] J. Z. Wang, Z. Du, R. Payattakool, S. Y. Philip, and C.-F. Chen, “A new method to measure the semantic similarity of go terms,” Bioinformatics, vol. 23, no. 10, 2007, pp. 1274–1281.

[21] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, “Text classification using machine learning techniques,” WSEAS Transaction on Computers, vol. 4, no. 8, 2005, pp. 966–974.

[22] E. Riloff and W. Lehnert, “Information extraction as a basis for high-precision text classification,” Transactions on Information Systems, vol. 12, no. 3, 1994, pp. 296–333.

[23] “MUC Data Sets,” 2014, URL: http://www-nlpir.nist.gov/related_projects/muc/muc_data/muc_data_index.html [accessed: 2014-01-06].

[24] N. L. Beebe and J. G. Clark, “Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results,” Digital Investigation, vol. 4, 2007, pp. 49–54.

[25] “20 Newsgroups,” 2014, URL: http://qwone.com/ jason/20Newsgroups/ [accessed: 2014-01-06].

Optimizing the Usability of Mobile Job Advertisements: A Multi-Method Approach

Stephan Böhm, Wolfgang Jäger, Sebastian Meurer, and Susanne J. Niklas
RheinMain University of Applied Sciences
Department of Media Management
Wiesbaden, Germany
e-mail: {stephan.boehm, wolfgang.jaeger}@hs-rm.de, meurer.sebastian@googlemail.com, susanne.niklas@gmx.de

Abstract— **Accessing the Web from mobile devices has become increasingly common even when searching for job information. Nowadays, most job board offerings are mobile-optimized. However, the search results often refer to job advertisements (ads) and external career pages that are not completely optimized for mobile access. For this reason, mobile users may be confronted with inadequate usability or a dissatisfactory user experience. In this context, the purpose of this study is to assess the usability of job ads posted on job portals to identify deficits and best practices. The analysis is based on an exemplary sample of job ads posted on a German job board. As a result, recommendations for a mobile-optimized design of job ads are presented.**

Keywords-Usability; User Experience; User Interfaces; Heuristic Evaluation; Mobile Recruiting.

I. INTRODUCTION

Mobile optimization of job advertisements is a requirement that follows from the increasing proliferation of smartphones and mobile media technologies today [1]. Recent studies point out that about 69 percent of all Internet users access the web using mobile devices. Within the target audience of 14- to 29-year-olds, as many as 80 percent are using the mobile web, underlining the general importance of this communication channel [2]. All in all, these developments will also noticeably affect recruiting processes, confronting the companies' personnel management with new challenges concerning the utilization of mobile technologies for personnel marketing and recruiting [3]. In this context, it is becoming more and more common to use these devices in order to retrieve job information as well. In Germany, 58 percent of all online job seekers are already accessing job information via mobile devices; in high-tech industries or the media sector, as many as 63 percent browse the mobile Internet for a new professional challenge [4]. Thus, mobile optimization is becoming essential in order to maintain reach among target groups and to keep up with the changing usage of media channels.

However, a mobile-optimized user experience in the area of mobile recruiting cannot be achieved by optimizing a single touch point within the recruiting funnel as shown in Figure 1. A potential candidate might be attracted by a search result in a job portal that is linked to a job ad. The job ad, in turn, can provide links to the company's career webpage or the applicant management system. All elements in this media chain supporting the recruiting funnel have to be adapted to the requirements of mobile devices to achieve a consistent mobile-optimized candidate experience.

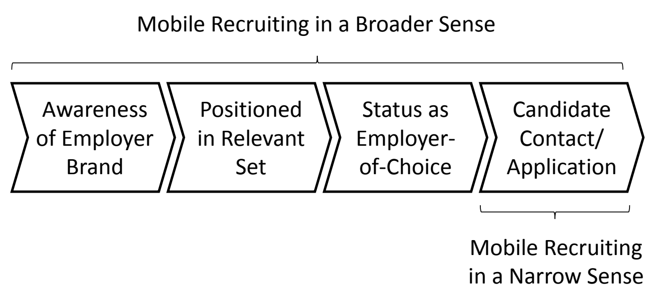


Figure 1. Mobile Recruiting in the "Recruiting Funnel" [5]

Nowadays, however, such a holistic mobile recruiting strategy is still rather rare. Companies often focus their activities on individual mobile recruiting elements. A prominent example is the incorrect usage of QR (Quick Response) codes in print media. QR codes can be used to facilitate mobile access to related online information. The QR code must, however, be linked to a mobile-friendly website, otherwise the user experience may be compromised. This can not only cause the user to abort the intended information retrieval, but also have a negative impact on the employer image [6]. Especially mobile recruiting in a narrow sense, i.e., the intended candidate application, cannot be achieved if an employer provides isolated mobile-optimized touch points. This discrepancy between the candidates' demand and the existing mobile recruiting offerings strongly supports the hypothesis that mobile optimization of the e-recruiting instruments along the recruiting funnel is still incomplete and requires improvement.

In this context, the paper at hand uses a multi-method approach to identify best practices and derive recommendations for the mobile optimization of job ads. Following this introduction, the research background and related work on mobile recruiting and mobile usability are the subject of Section II. The research methodology of this study is described in Section III before the key findings of this study are presented in Section IV. Based on the findings of the study, some management implications and recommendations for mobile-optimized job ads are derived in Section V. This paper is concluded by a discussion of the study's limitations and an outlook on future work in Section VI.

II. Research Background

Internet job search and job ads have been subject to research for more than a decade. While early research focused on the impact of Internet-based job search on labor markets in general [e.g., 7, 8], more recent work has analyzed online job seeker behavior and job ads in more detail [9, 10]. However, this research was conducted with regard to traditional, desktop-based access to the Internet by mainly focusing on economic and managerial aspects. Only a few research papers have been published on mobile job seeker behavior and the requirements for job ads in the context of mobile recruiting [3, 5]. For this reason, the research background and related work on mobile recruiting and mobile-optimization of job ads as well as the research gap to be closed by this paper are described below.

A. *Mobile Recruiting and Job Search*

According to a multi-year study on mobile recruiting in Germany [3], HR (Human Resources) managers attribute a growing relevance to mobile devices in personnel recruiting. In the latest study, conducted in 2013, 97 percent of the participating HR managers stated that addressing potential candidates via mobile devices is becoming increasingly important and almost as many expressed particular interest in the mobile recruiting topic. In addition, most participating HR managers (more than 90 percent) were quoted as being familiar with the main mobile recruiting tools and would be generally willing to provide major mobile recruiting content such as career-relevant information as well as job ads. The proportion of companies and organizations actively using mobile recruiting technologies and applications rose from 8 percent in 2009 to 25 percent in 2011, and 45 percent in 2013. A mobile optimized career website is offered by 26 percent of the companies. Altogether, the application of various mobile recruiting tools increased significantly from 2009 to 2013 [3, 11]. This observation is supported by a study of the German industry association BITKOM that was conducted at the end of 2012. According to this study, 24 percent of all German companies already offered a mobile optimized career website, followed by 17 percent with company-owned iPhone apps [4]. A study focusing on large enterprises in Germany revealed that as many as 80 percent of the companies provide a mobile career website and about 30 percent have a mobile career app [12].

A 2012 study analyzed German job seeker behavior and intentions in this area. At that time, only 6.4 percent of the respondents stated that they had already applied for a job using a smartphone or tablet. This is not a result of a lack of interest. In the same study, 30.2 percent of the respondents stated that they expect an attractive employer to support such a mobile job application [5]. As shown in Figure 2, the aforementioned mobile recruiting study from 2013 [11] also revealed substantial differences in the utilization of mobile recruiting technologies.

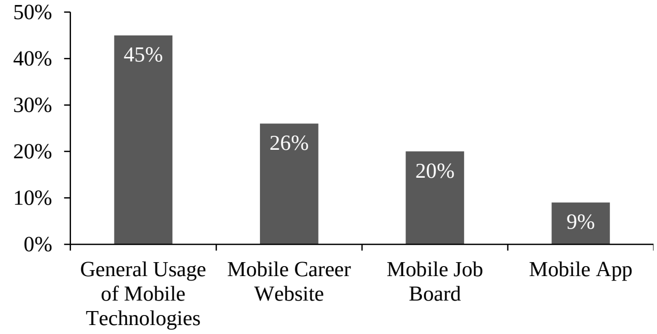


Figure 2. Usage of Mobile Recruiting in Germany 2013 [11]

Whereas nearly half of the participating HR managers in the study stated that they use mobile technologies in the context of their recruiting activities, not more than 26 percent operate a mobile career website. Only 20 percent offer a mobile job board and less than ten percent allow potential job seekers to apply directly via the mobile device. Against this background, it can be assumed that the industry is still far from achieving complete mobile-optimization of the recruiting process. As a result, candidates using smartphones might be confronted with a non-consistent mobile user experience and drawbacks because of media disruptions.

From the candidates' perspective, job portals are the preferred entry point and the most popular source of information for a mobile job search. The aforementioned study [5] on German job seeker behavior has revealed that the most frequently used sources of job ads on mobile devices are job portal websites, social media/business networks, search engines like Google, job portal apps, and employer career websites as shown in Figure 3.

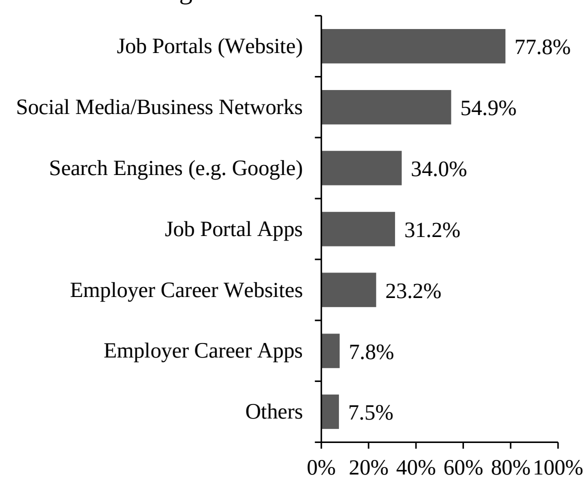


Figure 3. Sources of Job Ads Used on Mobile Devices 2012 [5]

For this reason, job portals and job ads are important entry points for mobile recruiting. Job portals complement the mobile recruiting activities of individual companies. Their providers aggregate job ads and career information across companies and sectors. In 2011, an analysis carried out for [13] the Apple App Store already identified ten mobile job board applications for the German market [14]. Employers who place job ads on job boards usually get a package for the online channels supported by the portal. When doing this, job board providers mobile-optimize access to their own portal functions, but may not alter the design of the job ads provided by a company. In that case, the search results of the job board can refer to a career website or a job ad that is not mobile-optimized. Thus, the mobile users may be confronted with inadequate usability or a dissatisfactory user experience.

As a result, all three interest groups are confronted with setbacks concerning their individual goals: The job seeker does not get the information he/she was looking for or has a poor user experience. Consequently, he/she probably decides to discontinue the app usage. The employer placing the job ad may experience a negative impact on the recruiting process, its employer branding, or may even lose a potential applicant. The job board provider, in turn, loses an app user, i.e., reach, which constitutes the basis of the job board business model. But even if a user does not directly discontinue the app usage, the design of the mobile ad and its content does play a major role concerning job ads' efficacy in terms of recall and retention [15].

However, regular usability guidelines for mobile websites cannot be applied directly to mobile job ads. Job ads provide very specific information within a focused area of application and thus require adapted criteria for usability analysis. But, despite the importance of these aspects and their high practical relevance, neither specially focused developer guidelines nor scientific research studies on mobile job advertising exist to date. To fill this research gap, the study at hand aims to identify deficits and best practices on a mobile-optimized job ad design, proposing a multi-method approach.

B. *Mobile Optimization of Job Ads*

Requirements for the design of mobile-optimized job ads can be found in guidelines for the user-interface design of mobile applications or mobile websites, e.g., the well-known *Best Practice Guidelines* of the World Wide Web Consortium (W3C) [16]. Here, recommendations are given regarding image format support, style sheet support, page weight, or color usage. However, two problems exist concerning the usage of such guidelines: firstly, the development as well as the improvement of modern smartphones are progressing at a furious rate. As a result, guidelines on principles for mobile development rapidly become outdated [17]. Secondly, those guidelines merely refer to technical capabilities and do not address the importance of different design aspects from the user perspective or usage context [18]. Some existing approaches, such as Nielsen's usability heuristic [19] or the adapted metric of the Microsoft Usability Guidelines (MUG, [20]), present a more holistic view on aspects influencing system usability. The MUG guidelines are based on the ISO 9241 usability definition, defining usability as the "Extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [13]. Besides structural evaluation in the form of heuristic analysis, user-oriented usability tests constitute an important evaluation method in order to measure efficiency, effectiveness as well as user satisfaction [13, 21]. User satisfaction can be measured by experience-based rating scales, product liking, or level of acceptance of the task solving effort [13].

As this study aims at giving practical recommendations for the design and development of mobile job ads, a multi-method approach with regard to both −structural evaluation as well as user based testing of usability aspects− will be adopted. In order to not just ensure success in terms of usability, but also in terms of a company's communication success, research on design aspects influencing the reception of job ads' content will be conducted additionally. The intended research approach will be described in the next section.

III. Methodology

Usability analysis can be classified in empirical and analytical methods. Empirical testing can comprise user and task observations of prototypes and final products by field or laboratory studies, including walk through and thinking-aloud analysis [19, 22, 23]. Heuristic evaluations, in turn, refer to assessment by a small group of evaluators according to a predefined set of usability guidelines or criteria [19]. As described above, mobile development often draws on technical guidelines and best practice standards, leading to the problems of being quickly outdated as well as not seeing the goal of overall usability concerning user satisfaction and usage acceptance [17, 18]. Heuristic usability evaluations however, by implementing a systematic inspection of user interface design aspects, enable the identification of usability problems to which special attention should be paid [19]. Here, two main methodologies are available for evaluation. Firstly, validator tools offer a standardized evaluation and in-depth analysis of websites, determining how well the site performs on mobile devices. Secondly, a heuristic evaluation can be carried out by looking at interface design in accordance with certain rules as listed in the guidelines. Here, a small number of evaluators (at least three) assess the compliance of a user-interface with usability principles, the so-called heuristics [19]. As presented in Figure 4, phase 1 of the study at hand implements two methods of usability evaluation for an exemplary set of mobile job ads: (1) A tool based usability evaluation by the *W3C mobileOK Checker* [24] and the *mobiReady testing tool* [25] validator. Both tools provide an overall value of "mobile fitness" as well as a detailed report on specific technical checks. (2) A heuristic analysis by evaluators, i.e., usability experts. The evaluation heuristic was defined by considering usability criteria of common standards, e.g., the W3C guidelines [16], the BBC Mobile Style Guide [26], the mobile applications of the MUG [20] or the Microsoft Mobile Design Guidelines [27].

In phase 2, empirical user testing is carried out to consider how users perceive mobile job ads and to identify usability issues and misconceptions from the user perspective [19]. Here, test subjects are asked to search for a job on the job board and to utilize presented job ads for this purpose (user walkthrough) by applying a thinking aloud approach for the analysis. This enables us to identify the job ads' major aberrations and drawbacks with which the user is confronted when attempting to achieve his or her goal and to evaluate design aspects within an actual usage context. Following this procedure, the test users will be asked to rate the likability of the observed job ads as well as to rank them in order of their preference to get a measure of the users' final satisfaction with the ads [13].

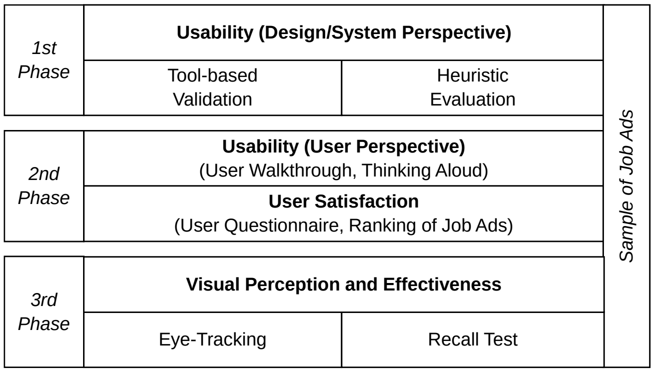


Figure 4. Study's Multi-method Approach

As mentioned above, the aim of the study is not only to evaluate pure usability aspects but also to assess the communication efficiency of the job ads. With this in mind, a third phase focusing on the visual perception and effectiveness of the job ads was conducted. For this purpose, one of the most advanced usability testing methods is the eye-tracking technique, which can be conducted directly on a mobile device (using head-mounted systems) or based on a simulation/representation of the design artefact on a desktop-based configuration. Thus, researchers are able to gain information on unconscious perception and information processing, which can be used to optimize user interfaces [28, 29]. As it has been shown that content related design aspects such as structure or visual design have a major influence on user perception and comprehension [30], these aspects were included in the study. To allow for aggregated group analysis and because the focus of this part of the study was on visual perception and not on user interaction, the study incorporates a desktop-based test configuration. The eye-tracking analysis was followed by recall tests on the user perception of the job ads' content. The users were asked to name companies, job titles or to recall employer brands in order to measure ad efficacy [15]. The combination of the results from the eye-tracking and recall testing is intended to gain recommendations for improving both, usability as well as communication effectiveness of the job ads. Aspects of the information quality [31] provided in the mobile-optimized job ad and its implications for the ease of finding appropriate job information in job portals are not analyzed but might be subject to future research.

IV. FINDINGS OF THE STUDY

The implementation of the multi-stage research approach to assess the usability of job ads was applied to a sample of 13 exemplary job ads from a German job board (partly mobile-optimized and non-mobile-optimized).

A. *Tool-based Validation and Heuristic Evaluation*

As a first step, the heuristic evaluation was applied to the sample. The evaluation was based on a heuristic that was developed by analyzing, consolidating, and adapting existing design guidelines to the specific requirements of this study as mentioned above. By doing this, e.g., the formerly advised maximum page size of maximum 20 kilobyte [16] was identified as no longer being up to date, since processing power and data transmission have improved tremendously [17]. Therefore, some more recent studies suggest that mobile pages should ideally not exceed 50 or maximum 100 kilobyte [32]. Other criteria refer to more detailed aspects like touch screen optimization, automatic redirects to mobile sites when accessed by mobile, the integration of inbuilt mobile functions like click-to-mail/-call, design aspects like font, contrast and images, as well as content related aspects concerning the appropriateness and relevance of information, e.g., job description, company, qualification or application.

As shown in Table I, a catalog with criteria subdivided into the categories access/navigation (ACN), design (DES), content (CON), and interactivity (INT) was derived. The catalog contained more than 30 criteria for the evaluation of the job ads and was intended to complement the tool-based assessment of "mobile fitness" mentioned in the preceding section. The tools calculate the mobile fitness as a percentage of mobile optimization. Likewise, each category of the heuristic evaluation was measured by assigning a percentage representing the extent to which the job ads comply with the criteria in the category as well as from an overall perspective.

TABLE I. AREAS OF HEURISTIC EVALUATION

Category		No. of. Criteria	Areas of Analysis (No. of Criteria)
ACN	Access/ Navigation	9	Mobile accessibility (3), use of mobile technologies (2), mobile optimized navigation (2), ease of access to additional sources (2)
DES	Design	12	Layout and structure (3), text and readability (3), mobile optimized embedding of media (6)
CON	Content	10	Corporate identity, appropriateness and relevance of employer and job information (8), contact channels
INT	Interactivity	5	Click-to-mail/-call, social media integration, locate job on map, option to apply via mobile device

The sample of job ads was assessed by eight evaluators using these heuristic criteria. An overall result was calculated based on the ratings of the two validation tools (VAL) and the consolidated heuristic evaluation (HEU). At this stage of the study, no weighting of the criteria, categories or types of tests was applied. This means the overall result was calculated as the arithmetic average of the partial results. Table II shows the average calculated across the job ads of the sample for the tool-based and the heuristic evaluation. The table also shows the lowest (Min.) and highest (Max.) rating as well as the difference (Diff.) between the highest and lowest ranking job ad within each category and for the overall result.

TABLE II. OVERALL RESULTS OF PHASE 1

	VAL	HEU	Heuristic (HEU) by Category				Over-all
			ACN	DES	CON	INT	
Avg.	27%	50%	43%	56%	77%	24%	38%
Min.	12%	38%	22%	37%	45%	3%	26%
Max.	54%	69%	74%	83%	98%	48%	48%
Diff.	42%	32%	51%	46%	53%	45%	22%

The key finding is that each of the examined job ads needs to be improved in order to provide an acceptable mobile user experience. None of the thirteen tested job ads achieves an overall rating of 50 percent. This is mainly caused by the dissatisfactory results for most of the job ads in terms of technical validation. However, the results provided by the validators and the heuristic rating as well as the resulting ranking (descending order from highest to lowest score representing the measured mobile usability) differ greatly in the majority of the cases as presented in Table III.

TABLE III. DETAILED RESULTS OF PHASE 1

	VAL		HEU	
Job ID	Score	Rank	Score	Rank
1	22%	7	45%	8
2	54%	1	38%	13
3	32%	5	50%	6
4	34%	4	41%	11
5	53%	2	43%	9
6	25%	6	42%	10
7	49%	3	47%	7
8	12%	11	61%	3
9	14%	10	54%	4
10	12%	11	39%	12
11	14%	9	63%	2
12	12%	13	54%	5
13	14%	8	69%	1

Many of the job ads achieve results between 10 and 20 percent in validation; only two of the job ads rated 50 percent or more. For the heuristic evaluation, all job ads reached 38 percent or more; six of them even achieved 50 percent or more. However, the results of both approaches did not correlate at all as shown in Figure 5. Moreover, as shown in Table III, the job ad with the highest score in the tool-based validation achieved the lowest score in the heuristic evaluation. This is due to the fact that the validators are somewhat outdated (feature phone focus) and do not consider the context (mobile recruiting) of use as the heuristic evaluation does. As a first result, a validation with the aforementioned validator tools cannot be recommended for mobile-optimization of mobile job ads.

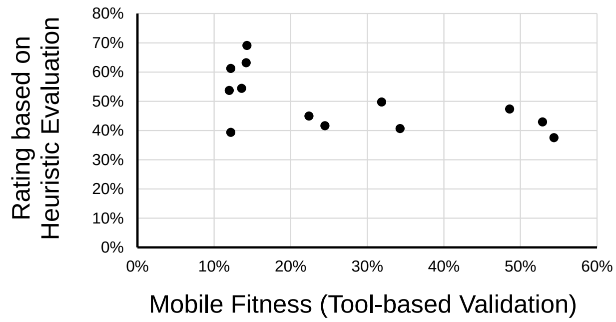


Figure 5. Usability Validation vs. Heuristic Evaluation

Focusing on the study results for the heuristic criteria in more detail, the weakest category of the heuristic criteria is “interactivity” as shown in Table II: not one of the examined job ads fulfilled half of the criteria. In contrast, the majority of the job ads achieve quite good results in the area of “content”, but this was also the category with the biggest difference between the lowest and highest ranking job ad. The job ads with above-average heuristic results lose their top positions in the overall rating because of their low score in validation. The top positions in the overall rating have only average scoring in heuristics, which is bolstered by a good validator score, possibly indicating a kind of trade-off between technical optimization and adoption of the technical capabilities of up-to-date smartphones. Overall, each of the analyzed job ads has plenty of room for improvement. In most cases, the technical “mobile fit” in terms of validation turns out to be poor. The performance of the mobile job ads in the areas of “content”, “design” and “navigation” is better, but far from good. Most notably, all of the tested job ads fail in the area of “interactivity”, where a good concept could really set a mobile job ad apart from the competitors.

B. User Walkthrough and User Satisfaction

The user walkthrough and the analysis of the user satisfaction with the job ads were conducted within a group of twelve test users. Each participant had to test between four and five job ads only, in order to prevent mental overstrain. As a result, most of the job ads were tested by four or five participants. This is a small number, due to the study’s limitations with regards to time and budget. In this context, it has to be considered that group sizes of five participants are often sufficient to find (on average) more than 80 percent of the usability problems [33] and that the purpose of this study is not to find all usability problems in the job ads but to compare their level of mobile-friendliness.

The walkthrough was based on some simple predefined tasks to standardize the usability testing. The participants had to find answers for questions like: By which date is the vacancy to be filled? What education is required? Who can be contacted for questions? Each test lasted an average of 15 minutes. At the beginning, each participant was given a brief introduction to the task. Once the participants were ready, the first job ad was presented on a smartphone and the participants were asked to complete the given tasks while thinking aloud. This procedure was performed for each job ad within the test users' subsample. All statements of usability problems were documented and consolidated after testing.

Most of the usability problems were caused by the fact that the information in the job ad was not adapted to the limitations of mobile devices ("Interaction elements are hidden, too small and so barely usable.", "Too much zooming and scrolling required to access the content") or the mobile usage context (e.g., "Registration process to apply via mobile is inconvenient.", "It is distracting to be forced to switch to the desktop site to apply for the job.")

The testing of job ads differs from the evaluation of other user interfaces. The design of the job ads varied widely from simple textual web pages with some text links to microsites with comprehensive information and a sophisticated navigation structure. Due to the aforementioned variation in complexity (text page vs. microsite), the number of usability problems did not seem to be an adequate indicator of the mobile fitness of the different job ads. This is why the number of interactions (e.g., touch gestures to click or scroll the screen) were measured instead. A lower average number of interactions to complete the tasks is interpreted as a higher level of mobile-optimization. Additionally, the participants were asked to make positive statements on the mobile-friendliness of the job ad when thinking aloud. The total number of positive user comments and the average number of interactions to solve the tasks for each job ad are presented in Table IV.

After completing the walkthrough, each of the study participants was asked to express his/her level of satisfaction with the provided usage experience by ranking the job ads within the tested subset (1 indicates the best ranking and the highest level of user satisfaction). Based on the rankings assigned to the job ads, an average ranking was calculated. If the participants had fewer than five job ads in their test sample the ranking was normalized. This was to ensure that the average ranking always varies between 1 and 5. The results of this ranking are also shown in Table IV and complete the findings of the second phase of the study.

The results presented in Table IV differ in detail. There is no direct relationship between the number of interactions, positive comments and the average user ranking. Some of the job ads received a high number of positive comments but also a relatively high average regarding the required number of interactions to complete the given tasks (e.g., job ad #12). The reason for this is that some of those job ads have a mobile-optimized design but also some usability problems hindering the users from efficiently completing the given task with a low number of interactions (e.g., buttons optimized for touch screens with unclear labeling). This supports the observation from the first phase that mobile-optimization cannot be achieved by technical transformation only but requires some additional adaptation of the content with regard to the mobile usage context. The results presented in Table IV also indicate that there is a strong correlation between the average number of interactions to complete a task and the user satisfaction as indicated by the average user ranking of the job ad.

TABLE IV. DETAILED RESULTS OF PHASE 2

Job ID	Number of Interactions (Average)	Positive Comments (Total)	User Ranking (1 to 5)
1	2.33	6	3.00
2	2.47	5	3.50
3	2.00	8	1.25
4	2.20	0	2.67
5	1.43	2	2.67
6	1.30	6	2.50
7	2.37	5	3.00
8	2.70	5	2.25
9	1.60	4	1.25
10	2.33	5	3.25
11	2.70	1	3.25
12	3.57	7	4.50
13	1.93	18	1.60

Figure 6 shows a scatter plot of these two variables. The Pearson correlation is 0.679 and significant at the 0.05 level (p=0.011) whereas there is no significant correlation between the number of positive comments and user satisfaction expressed by the user ranking (Pearson r= -0.312, p=0.299).

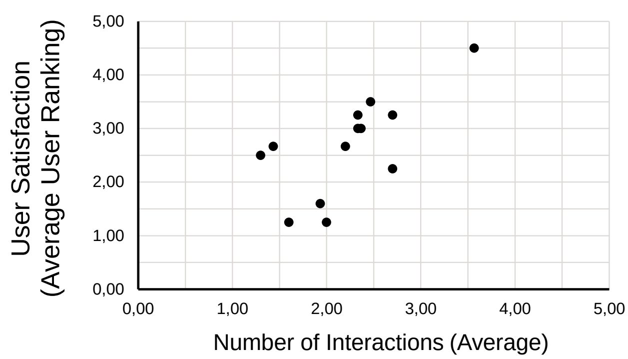


Figure 6. User Satisfaction vs. Number of Interactions

The findings of this phase suggest that when mobile-optimizing job ads, the minimization of the required number of interactions should be focused on. This requirement is consistent with the statement of [34] that "ease of use is paramount", i.e., that mobile design has to cope with distractions, background noise and interruptions as well as device constraints that require a minimization of user input.

C. Eye-Tracking and Recall Testing

The third phase of the study started with an eye-tracking analysis. All 13 jobs were tested by 15 participants. As a prerequisite, slides with screenshots of the job ads were produced and merged into a slideshow project. The slides contained the area of the smartphone screen, which can be seen on a smartphone (iPhone 4S) without any scrolling. No interaction with the screen was considered in this testing. The participants received a short briefing and were asked to identify the employer, job title and elements that distinguish the job ads from others. All the slides were presented to the participants with a duration of 15 seconds for each. The presentation of the ads was carried out on a desktop-based eye-tracking system to allow for group consolidation. Thus, the smartphone screens were not presented in the original size on the desktop screen. The size was adjusted for the viewing distance of approximately 60 cm between desktop screen and participant to simulate the smartphone user experience.

Based on the eye-tracking analysis of the slideshows, heatmaps were generated as shown in Figure 7. The heatmaps represent the consolidated group data by visualizing the aggregated eye fixations of all participants for the job ad. Hotspots within those maps are interpreted as areas of high attention and interest.

Figure 7. Eye-Tracking Analysis Example

The aggregated heatmaps were used in our study to derive qualitative findings on how to improve the visual design of the job ads and how to optimize the positioning of the different job ad elements on the smartphone screen.

To gain some additional quantitative data from the tests, the generation of heatmaps was complemented by an area of interest (AOI) analysis [35]. Here, the eye-tracking software was used to define areas within the screen where the employer logo and the job description were located for each job ad as shown in Figure 7. The eye-tracking data collects data such as the time to first fixation, the number of fixations, or the fixation duration for each of those defined areas.

The results of the third phase of the study are shown in Table V. Each job ad is listed with the total time to first fixation for the employer logo and the job title that were defined as areas of interest and analyzed with the eye-tracking software. In some of the screenshot slides, a logo was not available (marked as n.a.) – either because the companies had decided not to use a logo or the logo was not visible without scrolling. Subsequent to eye-tracking, a cued recall test was conducted with all the participants. The participants were asked if they remembered the employers and the job titles of the presented job ads. The percentages of the participants who were able to recall the employers and the job titles are shown in Table V. The recall rate for the employer was very low for job ads without the integration of a logo. Therefore, the findings indicate that an employer logo needs to be integrated in the job ads and has to be visible without scrolling.

TABLE V. DETAILED RESULTS OF PHASE 3

	Employer (Logo)		Job Title	
Job ID	**TtFF [s]**	**Recall**	**TtFF [s]**	**Recall**
1	4.074	47%	3.009	7%
2	2.239	53%	3.194	20%
3	2.837	33%	4.147	7%
4	1.005	27%	3.160	0%
5	n.a.	7%	2.825	0%
6	n.a.	7%	0.485	0%
7	0.597	67%	3.884	0%
8	2.135	60%	1.123	7%
9	3.654	67%	2.704	7%
10	n.a.	7%	2.363	0%
11	2.768	80%	3.558	20%
12	0.919	27%	2.338	33%
13	3.565	20%	0.557	7%

Note: TtFF = Time to First Fixation

However, there was no significant correlation between the time to first fixation and the recall rates. A possible explanation might be that the job ad design was not systematically modified but based on a random sample and other design or content factors influenced the recall results. Another source of bias might be the attitudes and preferences of the study participants. In retrospect, it was found to be inappropriate for the recall tests to use a random sample of real job ads. This is because the selected employers differ significantly in their brand awareness, which means that the recall results depended more on the brand awareness than the mobile-optimized positioning of the employer logo on the screen. Additionally, the popularity and understandability as well as the extent to which the personal interests of the study participants matched the job titles varied too much within the random sample (e.g., bus driver vs. public relations trainee for healthcare communication).

V. Conclusions and Implications

The objective of this research was to use a multi-method approach to gain some best practice guidelines for the mobile-optimized design of a job ad. As a first step, a random sample of real job ads from a job board were analyzed. The study at hand has some limitations (e.g., small sample size and small number of test users). However, the study revealed some methodological problems and areas for further research with regards to mobile usability of job ads and mobile usability in general:

a) First of all there is a *lack of applicable and commonly accepted mobile usability criteria/metrics*. Some of the existing metrics focus too much on feature phone characteristics and do not incorporate the capabilities of modern smartphones.

b) As a result the *popular validators for mobile fitness are outdated* as well and not appropriate for the content presented on modern smartphones. In addition, based on the results of the study, it may be doubted that a rating of mobile-optimization gained by the usage of those validator tools is meaningful at all. Mobile-optimization requires a high level of adaptation of the user interface design and the presented content towards the intended mobile usage context. This aim cannot be achieved by just adapting the code to comply with the constraints of mobile devices. Thus, the scope of the validator tools is rather limited as far as the identification of technically related usability problems is concerned.

c) The job ad rankings varied very much between the different methods used for the analysis. This is because *mobile-friendliness is a multi-dimensional construct* with some trade-offs between the individual dimensions that need to be optimized for mobile usage. Thus, it is not sufficient to rely on one single method when aiming for mobile-optimization. Usability aspects have to be analyzed in line with the functional requirements and other limitations such as user habits and expectations as well as corporate design restrictions.

d) The study has shown that quantitative analysis is of limited use in finding a generalizable mobile-optimized design pattern. This underlines the *necessity of obtaining qualitative user feedback* and of involving users from the target group in the development process of the user interface design.

Besides the conclusions discussed above, the study revealed some first learnings on the design requirements of mobile-optimized job ads. These findings are still far from providing a consistent and scientifically proven design guide. However, the following recommendation can be a starting point to optimize the job ad design and the resulting candidate experience within the mobile recruiting process.

A first recommendation is based on the observation that the feature phone-based optimization approach is not sufficient to optimize user experience on modern mobile devices:

- *Cross-platform compatibility* is a design prerequisite due to the high fragmentation of mobile devices that accompanies continuous technical progress. There are different types of devices (e.g., feature phones, smartphones, tablets) and even the devices of the same type can differ significantly with regard to their technical capabilities and constraints (e.g., screen size). Thus, job ads should be realized by using common web technologies to provide cross-platform compatibility. Responsive web design technologies and frameworks can help to optimize the user experience across platforms and devices.

The following recommendations are based on the quantitative findings as well as the qualitative feedback gained in the multi-method study presented above:

- *Company name and logo are key visuals* and thus need to be placed at the top of the page to be visible without any scrolling. Especially companies with strong employer brands can use these elements to directly draw attention to the job offering.
- *Job title, start date, requirements and location* are other important elements that need to be positioned in the upper screen area. These elements should be accessible without the need to scroll, browse through the content, or navigate to linked pages.
- *Sparse use of pictures and graphical elements* to enable the users to focus on the key elements of the job ads even when the job ad is viewed in a mobile usage context with a high level of distractions, background noise and interruptions.
- *Single column formatting* is required to prevent users from having to scroll vertically and horizontally to browse through the content.
- *Supplementary information* on job details, company and the location can be provided by linking to external but mobile-optimized content. Content such as mobile video should be preferred to excessively long texts that will not be read in mobile usage contexts characterized by short attention spans.
- *Touch-optimized buttons* should be used all over the job ad instead of text- and image-based links.
- *Contact details* should be presented on the first page of the job ad and provide direct access to the appropriate features of the mobile devices without the need to re-enter the contact details.

Figure 8 shows an example design of a job ad that is based on the recommendations above. As mentioned earlier, this generic design has to be adapted to the company's needs (e.g., corporate design).

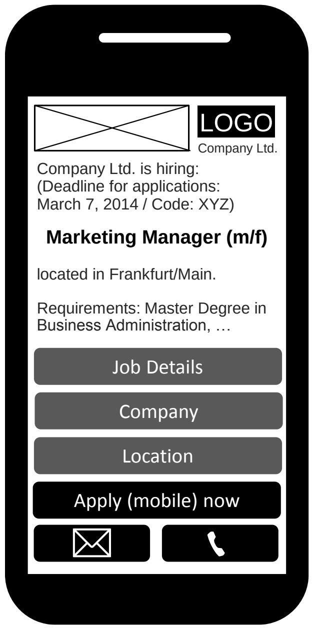


Figure 8. Mobile-optimized Job Ad Example

Usage of the more advanced features to apply for a job via mobile device as shown in Figure 8 ("Apply now") will depend on the appropriate preparation and transformation of the underling recruiting processes and backend systems as already discussed in Section I.

VI. Limitations and Future Work

Further research is required to gain more insights on the impact of the suggested design recommendations on user experience and user acceptance. As mentioned before, the usage of a random sample of job ads may have biased the study results and also have limited the applicability of qualitative methods.

For this reason, subsequent studies should focus on a more systematical variation and combination of design elements. For this purpose, A/B testing could be applied. Within the so-called A/B testing, various user interface design alternatives are analyzed to obtain design recommendations, i.e., design best practices. To gain insights, single design attributes (like typeface or button design) are varied and the resulting design variants are evaluated against each other. Analysis of the different versions can either be done in a live setting by, e.g., tracking conversion rates of design alternatives or within an experimental laboratory environment [36]. An experimental setup with a combination of A/B testing with other types of analysis, e.g., eye-tracking or recall testing, could then be used to gain a better understanding of user interaction and visual perception of the presented job ad [28].

Another important research question is the analysis of the reasons why the job ad providers have not adapted their ads to mobile environments yet. Reasons could be manifold, e.g., technical, lack of budget, or insufficient knowledge of user needs and usage shift to mobile devices. As the focus of the study at hand was to derive recommendations on mobile-optimization, this topic could not be investigated here but may be the subject of further research.

References

[1] S. Böhm, S. Niklas, and W. Jäger, "A Multi-Method Approach to Assessing the Usability of Mobile Job Advertisements (Working Paper)," Proceedings of CENTRIC 2013 the 6th International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC), 2013, pp. 108–111.

[2] Accenture, Mobile Web Watch 2012. Special edition: Germany, Austria, Switzerland, 2012.

[3] S. Böhm and S. J. Niklas, "Mobile Recruiting: Insights from a survey among German HR managers," Proceedings of the 50th ACM SIGMIS annual conference on Computer and People Research (ACM SIGMIS CPR'12), Milwaukee, Wisconsin, 2012, pp. 117–122.

[4] BITKOM, 2012, Mitarbeitersuche per Smartphone. http://www.bitkom.org/de/presse/74532_74308.aspx. Accessed 18 February 2013.

[5] S. Böhm, "Behavior and expectations of mobile job seekers," Proceedings of the 51st SIGMIS annual conference on Computers and people research (ACM SIGMIS CPR'13), Cincinnati, Ohio, 2013, pp. 105–110.

[6] S. Böhm and S. J. Niklas, "Mobile Tagging in the German Market: A Comparative Study on User and Non-User Characteristics," International Journal of Business and Management Studies 4, 2, 2012, pp. 161–170.

[7] D. H. Autor, "Wiring the Labor Market," Journal of Economic Perspectives 15, 1, 2001, pp. 25–40.

[8] P. Kuhn and M. Skuterud, "Job search methods: Internet versus traditional," Monthly Labor Review 123, 10, 2000, pp. 3–11.

[9] B. Stevenson, 2008, The Internet and Job Search. http://www.nber.org/papers/w13886.

[10] V. Brenčič and J. B. Norris, "Do employers change job offers in their online job ads to facilitate search?" Economics Letters 108, 1, 2010, pp. 46–48.

[11] S. Meurer, W. Jäger, and S. Böhm, 2014, Mobile Recruiting 2013. Eine empirische Untersuchung zur Bewerberansprache über mobile Endgeräte. http://www.djm.de/mr2013/mobile-recruiting-2013_bericht.pdf. Accessed 5 March 2014.

[12] JobStairs, Bewerber und Unternehmen setzen schon heute auf mobile Jobangebote, Bad Nauheim, 2012.

[13] ISO – International Organization for Standardization, 1998, Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability, 9241-11.

[14] S. Böhm, W. Jäger, and S. J. Niklas, "Mobile Applikationen im Recruiting und Personalmarketing," Wirtschaftsinformatik und Management 3, 4, 2011, pp. 14–22.

[15] M. de Sa, V. Navalpakkam, and E. F. Churchill, "Mobile Advertising: Evaluating the effects of animation, user and content relevance," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2013, pp. 2487–2496.

[16] W3C, 2008, Mobile Web Best Practices 1.0. W3C (World Wide Web Consortium). http://www.w3.org/TR/mobile-bp/. Accessed 12 September 2011.

[17] E. English, 2012, W3C Mobile OK Checker and Mobile Web Best Practices – a bit outdated? http://www.pijnz.com/blog/w3c-mobile-ok-checker-and-mobile-web-best-practices-a-bit-outdated/. Accessed 28 May 2013.

[18] A. P. Massey, V. Khatri, and V. Ramesh, "From the Web to the Wireless Web: Technology Readiness and Usability," Proceedings of the 38th Hawaii International Conference on System Sciences, 2005, pp. 1–10.

[19] J. Nielsen, Usability Engineering. Academic Press, Boston, 1993.

[20] V. Venkatesh and V. Ramesh, "Web and wireless site usability. Understanding differences and modeling use," MISQ 30, 1, 2006, pp. 181–206.

[21] A. R. Hevner, S. T. March, and J. Park, "Design Science in Information Systems Research," MISQ 28, 1, 2004, pp. 75–105.

[22] R. Agarwal and V. Venkatesh, "Assessing a firm's Web presence: A heuristic evaluation procedure for the measurement of usability," Information Systems Research 13, 2, 2002, pp. 168–186.

[23] C. Coursaris and D. J. Kim, "A meta-analytical review of empirical mobile usability study," Journal of Usability Studies 6, 3, 2011, pp. 117–171.

[24] W3C, 2013, W3C mobileOK Checker. http://validator.w3.org/mobile. Accessed 21 August 2013.

[25] mobiReady, 2013, mobiReady testing tool. http://ready.mobi/launch.jsp?locale=en_EN. Accessed 21 August 2013.

[26] B. Guyer, R. Puustinen, and C. D. D. Urschabch, BBC Mobile Style Guide I.I – Global visual language for the mobile Web, 2009.

[27] Microsoft, 2010, Design Guidelines. http://msdn.microsoft.com/en-us/library/bb158602.aspx. Accessed 29 May 2013.

[28] P. Chynał, J. M. Szymański, and J. Sobecki, "Using eye-tracking in a mobile applications usability testing," Proceedings of the 4th Asian conference on Intelligent Information and Database Systems (ACIIDS'12), 2012, pp. 178–186.

[29] C. C. Seix, M. S. Veloso, and J. J. R. Soler, "Towards the validation of a method for quantitative mobile usability testing based on desktop eyetracking," Proceedings of the 13th International Conference on Interacción Persona-Ordenador (INTERACCION '12), 2012.

[30] E. Cuddihy and J. H. Spyridakis, "The effect of visual design and placement of intra-article navigation schemes on reading comprehension and website user perceptions," Computers in Human Behavior 28, 4, 2012, pp. 1399–1409.

[31] Shirlee-ann Knight and Janice Burn, "Developing a framework for assessing information quality on the World Wide Web," Informing Science Journal 8, 2005, pp. 159–172.

[32] Virginia Commonwealth University, 2011, VCU Web Standards & Guidelines. Mobile Sites. http://www.webstandards.vcu.edu/mobile/. Accessed 2 June 2013.

[33] L. Faulkner, "Beyond the five-user assumption: benefits of increased sample sizes in usability testing," Behav Res Methods Instrum Comput 35, 3, 2003, pp. 379–383.

[34] S. Nadav and J. Braiterman, "Design Sketch: The Context of Mobile Interaction.," MobileHCI '07 Proceedings of the 9th international conference on Human computer interaction with mobile devices and services.

[35] M. Manhartsberger and N. Zellhofer, "Eye tracking in usability research. What users really see," Proceedings of the 1st Usability Symposium, HCI&UE Workgroup, pp. 141–152.

[36] M. Hampson, 2010, A/B testing. http://www.bbc.co.uk/blogs/webdeveloper/2010/01/ab-testing.shtml. Accessed 29 May 2013.

MobileSage – A Prototype Based Case Study for Delivering Context-Aware, Accessible, and Personalized On-Demand Help Content

Till Halbach & Trenton Schulz

Norwegian Computing Center
Oslo, Norway
Email: {till.halbach,trenton.schulz}@nr.no

Abstract—**We present the system design decisions for the MobileSage prototypes, a service for the on-demand delivery of multimodal and accessible help content to anyone in general and seniors in particular. Findings from user-centered research formed the system requirements, as well as design considerations and decisions. The design of the system also includes availability, relevance, accessibility, conciseness, and comprehensiveness of multimodal content. The prototypes have been evaluated in multiple user trials with good results, showing the participants' high appreciation of such a service and a moderate to high degree of satisfaction with the prototypes.**

***Keywords*–Mobile; smartphone; application; assistance; guidance; help on demand; personalization; adaptive; accessible; usable; multimodal; context; location aware; content management, Ambient Assisted Living, AAL JP.**

I. Introduction

Previous work [1] has documented that the ever increasing number of machines and devices surrounding us in our everyday lives poses a challenge to many, and elderly persons in particular. Many senior citizens view solutions based on Information and Communication Technology (ICT), such as ticket machines and web services, with anxiety. These solutions add to an experienced rise of complexity in their life and, instead of being task enablers, hinder their ability to accomplish the task. At the same time, modern elderly – here defined as people aged 65 and older – live longer, are healthier, more active, mobile, independent, and more demanding customers than ever [2]. Estimates mention approximately 87 million elderly in Europe [3]. They are increasingly looking for useful, user-friendly, and personalized ICT services that add value to their active and mobile life, that help them solve everyday tasks instead of complicating them, and they also desire services that can help them stay active despite various impairments.

MobileSage provides a timely approach and solution to these problems. Its main idea is to provide assistance to virtually anybody, but particularly elderly, in solving everyday tasks. Someone with interest in assisting an individual can register instructional help content, potentially split up into several steps, in multiple languages and multiple modalities, such as images, text, video, audio, and accompanied by geo-locational data. The content can then be accessed upon demand; for instance, upon scanning a QR code, scanning an NFC tag, through plain text search, and through location search.

This work is based on a previous conference article [1]. The article has been extended with related research and the description and results from two more user evaluations. Its contributions include:

- an overview of related solutions and research projects,
- the discussion of technical considerations and decisions regarding system architecture, functionality, and design,
- the results from a three user evaluations, and
- a method for efficient user involvement throughout IT projects.

The article is organized as follows: First, we give an overview of MobileSage and related research, before presenting the system architecture and all main components. Next, we lay out the requirements for these components and their consequences for system design and content production. Then, we present and discuss the three user evaluations in Norway and their results. Finally, we present conclusions from the project.

II. MobileSage Overview

MobileSage stands for *S*ituated *A*daptive *G*uidance for the Mobile *E*lderly. The name refers to the project and its main deliverable, a mobile app. The consortium was a mix of software companies, research institutions, industry partners, and end-user organizations from Norway, Romania, and Spain. The project lasted from July 2010 through April 2014 and was funded by the Ambient Assisted Living Joint Programme (AAL JP).

This program had been created to fund ICT-based innovation projects targeting elderly individuals [4]. The main goal of AAL is to improve "the quality of life, autonomy, participation in social life, skills, and employability of older people", while service delivery enhancement and care costs reduction are secondary targets. Solutions in form of products, services, and systems should aim at a market introduction within the next 2-3 years after the project end. The third AAL JP Call was issued in April 2010 with the title "ICT-based Solutions for Advancement of Older Persons' Independence and Participation in the 'Self-Serve Society"' [5].

The Call also specifies the target groups that the projects should be tailored for. The *primary* end-user group, consisting of elderly individuals with or without impairments (motor, perception, cognition), might have little or no familiarity with technology. Ideally, these individuals have the explicit wish to remain active members in the digital society.

Then, there are family members and care persons, both of whom stand close to the primary end-users, and which add up to the *secondary* user group. As amateurs, they may have an interest in producing or finding help content relevant for the primary users.

The group of *tertiary* end-users denotes NGOs teaching digital literacy to the elderly, public actors like city authorities, and private actors such as transport companies and machine vendors. As with the secondary user group, tertiary users may want to produce, modify, or make help content available to the primary users.

III. RELATED RESEARCH

As part of the preparation of the project, an effort mapping updated information about state of the art in the field was carried out[1]. The study revealed that there are few, if any, existing solutions as described in the MobileSage project. Especially rare are related projects and solutions of this particular kind directed towards elderly and impaired users. However, the study showed that there is a number of recent and ongoing international projects related in various ways to the topics addressed by MobileSage; they are discussed subsequently.

A number of international projects fall within the same *scope*.

The (ongoing) APSIS4ALL project – Accessible Personalised Services In Public Digital Terminals for all – deals with personalizing public digital terminals such as ATMs and ticket machines [7]. An adaptive interface and personalized interaction is achieved by the human communicating with the machine through a smartphone.

The ASK-IT project – Ambient Intelligence System of Agents for Knowledge-based and Integrated Services for Mobility Impaired users – developed a framework that provides intelligent agents for service provision and search for suitable semantics [8]. It also allows to personalize service and content, and personalizes user interfaces. The project focused on daily life and travel scenarios.

The ongoing MyUI project – Mainstreaming Accessibility through Synergistic User Modeling and Adaptability – addresses specific user needs towards ICT products in general through adaptive personalized interfaces [9]. Its ontology-based framework collects user and context information in real-time during use in order to establish an evolving user model to which user interfaces of various personal applications can adapt, by means of empirically based design patterns, as such data is shared among services.

GUIDE – Gentle User Interfaces for Elderly People – is an ongoing project to design tools and aids for developers to efficiently integrate personalization, user-friendly interaction, and accessibility features into applications [10]. Here, the focus is on Web, TV, and middleware like set-top boxes.

The goal of the DIADEM project – Delivering Inclusive Access for Disabled or Elderly Members of the Community – was to make electronic and online forms adaptable to the cognitive skills of the user [11]. It employed an expert system that monitored the user actions and behavior; it personalized and tailored the user interface based on this observation.

The ongoing GPII Project – Global Public Inclusive Infrastructure – is building a framework that stores universal user profiles in the cloud [12]. The profile can be accessed to adapt the user interface of any device to a user's needs and preferences. Also, a marketplace will be established for developers to share assistive-technology tools for accessible services and content.

The following research projects are related to *user interface* aspects of MobileSage.

The aim of SNAPI was to develop a data format for the storage of user profiles, with a focus on smartcards [13]. Personal preferences and other settings could be stored in the user profile to personalize the human computer interface of public digital terminals and adapt it to the user's needs. The format has recently become a CEN European standard [14].

The objective of the ongoing GoldUI project – Adaptive embedded human interfaces designed for older people – is to offer the elderly a number of cloud services that are deemed useful in the everyday life [15]. These services can be accessed through a variety of platforms, including telephone, smartphone, tablet PC, TV, and radio. The services include traditional broadcast services like news syndication, music playback, and weather forecast, combined with services like calendar, task lists, online shopping, and social media.

The following research projects address transportation. The ongoing WayFiS project – Way Finding for Seniors – addresses the topic of travel challenges [16]. The project plans include both a web based pre-planning service and a mobile application. Route calculations take account an individual's desired physical activity, nutrition needs, necessary facilities along the route, and disease restrictions, while trying to avoid inaccessible places.

Mediate – Methodology for Describing the Accessibility of Transport in Europe – is a completed project that has developed criteria and tools to measure accessibility in public transport, including accessible ticketing and information systems [17]. Also, a database has been erected with accessibility information of public transport system in Europe.

In the Access2all project – Mobility Schemes Ensuring Accessibility of Public Transport for All Users – the accessibility of public transport was considered [18]. Concrete outcomes of the project were a number of guidelines, recommendations, roadmaps, and new research initiatives.

The AmbienNet – Ambient Intelligence Supporting Navigation for People with Disabilities – created an indoor location system based on intelligent infrastructure and a sensor network has been developed [19]. The resulting navigation system could assist users with and without disabilities.

The following projects address *multimodality*.

[1]This section was first presented in a conference [6].

The ongoing HAPTIMAP project – Haptic, Audio, and Visual Interfaces for Maps and Location Based Services – develop a cross-platform toolkit for the design of user interfaces incorporating haptic, audio, and video input and output that goes beyond guidelines and checklists [20]. The project focuses on the retrieval, storing, and manipulation of geographic data.

The MAPPED project – Mobilisation and Accessibility Planning for People with Disabilities – developed a mobile application that provides accessibility information on buildings and traveling on buses or trains [21]. The service employed localization techniques to increase the relevance of the information.

The HearMeFeelMe project – Compensating for Eyesight with Mobile Technology – aimed at replacing visual and textual information with audio, combined with touch-based interfaces for information access [22]. The project included a mobile application and a system for object mapping and tracking in indoor environments. The application employed near-field technology to gather information about items to buy, such as food and medication [23].

IV. MobileSage Components

The MobileSage prototype consists of two components: the Help-on-Demand (HoD) mobile application and the Content Management Service (CMS). Figure 1 shows the overall architecture.

A. *Help-on-Demand Service*

The HoD application is a personal agent, i.e., a thick-client application running on a smartphone. It is built up in a service-oriented manner, as shown in Figure 1. The user interacts with the Dialog Manager through the User Interface. The Dialog Manager uses information from the Profile Service, which takes care of the user profile. The user profile stores personal preferences and usage patterns. User behavior and User Interface events are logged and analyzed by the Personalization Service, upon which the user profile is re-adjusted [24–28].

The Dialog Manager is in contact with the Reasoning Service to help determine the user's context. Reasoning makes use of network services such as Media Service, Search Service, and the Content Service. The Reasoning Service gets help from the Localization Service, which can determine the user's location based on technologies like A-GPS, WLAN, GSM/GPRS, NFC, and triangulation methods.

The HoD Service requests any content from the CMS upon initiation of the user.

B. *Content Management Service*

The CMS is a cloud service running on a web server. Content producers interact with the service's Dialog Manager, which in turn controls the User Interface on a User Agent like a web browser. The logic for handling the multimodal content lies in the Content Manager, which has a modular design to be able to add additional modalities in a simple way. The prototype supports the modules Video (with or without captions), Animation, Image, Audio, Text, and Formatted Text (basically simplified HTML). The content is stored by the Content Service. It is also possible to refer to content located elsewhere (e.g., from other video services) through proper hyperlinking and HTML redirects.

There is no limitation to the kind of content that can be served. This includes manuals, usage instructions, descriptions of travel routes, and geographical points of interest. We anticipate that machine makers and service providers will generate most of the content. For instance, a particular vendor might provide manuals for their ticket machines, or the railway operator that runs these machines might do so. Even a municipality might be interested in producing such help content as a special service for their citizens and visitors. Other interested parties are expected to add content to the CMS to fill in the gaps left behind by product makers and service providers, like caregivers, relatives, and friends, as they are likely to have a direct interest in helping particular individuals. Finally, there is nothing that prevents users of the HoD from producing and making help content available themselves.

V. User and System Requirements

The following sections address the formulation of the requirements and constraints for the system design.

A. *Requirements for Primary Users*

The derivation of the requirements of primary users is split into the gathering of the users' expectations towards the system (user needs analysis), and the collection of user requirements. The system requirements were derived from the latter.

1) User Needs Analysis: Focus group work was conducted in the three countries Norway, Romania, and Spain to find the needs of primary users [29]. The focus groups had 39 participants and represented a broad range of parameters, including age (48 to 96), gender (24 female vs. 15 male), disabilities (sensory and cognitive impairments), nationality (four foreigners), and ICT experience and usage. Two scenarios were presented to the participants: a traveller with reduced vision in a foreign country who was not proficient in the language, and an elderly lady at home trying to understand how to use an electric household appliance.

The focus groups' results show that "modern elderly persons" are a heterogeneous group with a wide range of – sometimes contradictory – needs and wishes. This applies also to the users' familiarity with ICT in general and mobile technology, which ranges from none to advanced users. However, it was possible to identify themes of functionality the solution should have [29]. More specifically, the solution should

- lead to higher independence of elderly people according to the help-for-self-help principle,
- increase a person's mobility and be usable for transportation and travel, including holidays and visits,
- be applicable in the home environment and throughout daily living,
- provide relevant, useful, context- and location-sensitive and multimodal (and hence accessible) assistance in an on-demand manner,

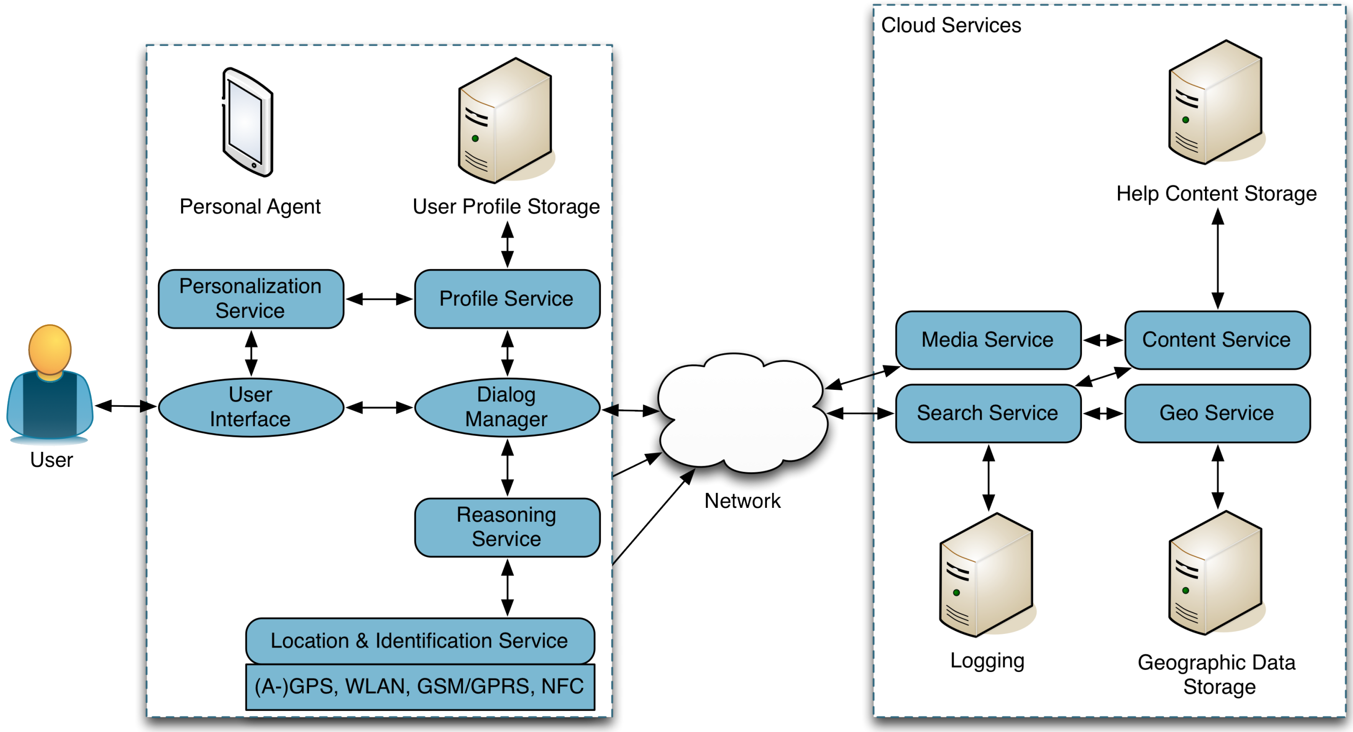


Figure 1. System architecture and major building blocks for HoD (left) and CMS (right)

- be accessible, user-friendly, designed for all, possible to personalize or customize, adaptive, social, and
- honor privacy, security, and trust matters.

2) User Requirements: The results from the user needs analysis were collected and formulated as user requirements [30]. The roughly 50 requirements mirror the expectations of primary users regarding HoD, but were extended to be valid for the CMS as well. The user requirements served as input to the process of formulating the first draft of the system requirements for the service's two components.

3) System Requirements: The technical requirements for the Help on Demand and Content Management services were derived from the user requirements.

The requirements specification for HoD has over 60 requirements [31], while the CMS specification contains only 40 [32]. Both address topics such as system functionality, user interface, and input and output matters. Also included are sections on the technology choice and mockup examples regarding the services' user interfaces.

B. Requirements For Secondary and Tertiary Users

MobileSage's focus is on primary users. Secondary and tertiary users have been accounted for by formulating a set of requirements representing the needs of the transport company participating in the project. These are as follows:

- It should be possible to identify one or several points of interests with a unique ID.
- There could be multiple help topics per ID.
- One topic could be presented in multiple languages.
- The service should support content hosted elsewhere ("upload once, available everywhere").
- It should be possible to edit help content in order to add locations, languages, and modalities.

VI. System Design

For the HoD service, a user profile lays the ground for personalization and adaption of the service. It contains the user's settings and preferences, such as font size, emergency number, accepted media types, and additional languages. Also other parameters are stored there, including usage log. This log is the basis for system adaptation. Screenshots of the HoD are shown in Figure 3.

Both primary and tertiary users have requested that it should be possible to associate content with specific locations or points of interest. However, it should also be possible to link certain content to several locations (e.g., "how to buy a ticket" could be valid for any ticket machine in a municipal area). Moreover, there are situations where several pieces of content are relevant at a single location (e.g., how to validate a ticket, arrival time of the next bus, or choosing the correct platform for departure).

These issues have been solved by the Content Item, see Figure 2. This uniquely identifiable item is a logical unit to gather content that is related to each other. Multiple locations in terms of latitude, longitude, and altitude can be linked to a

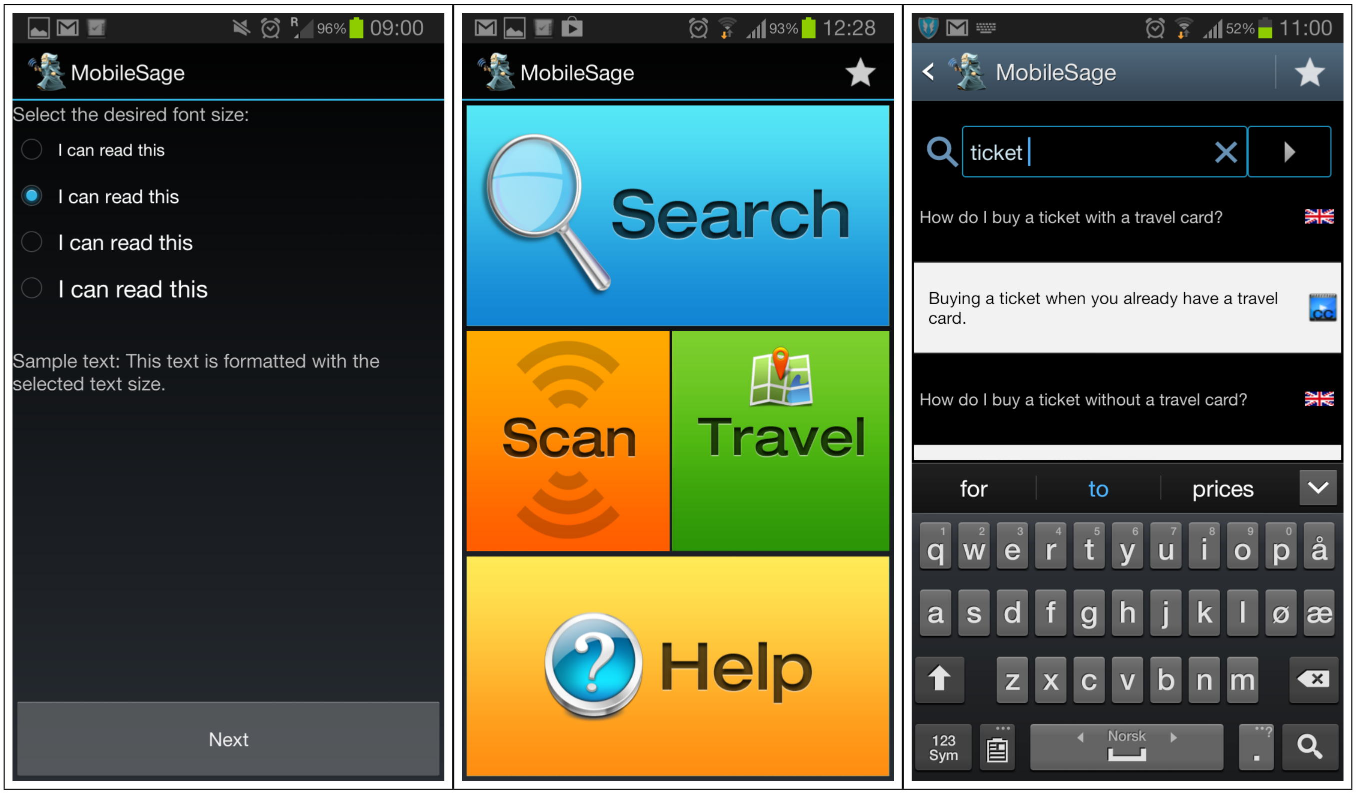


Figure 3. Screenshots of the Help-on-Demand application (V1.0): set up (left), home screen (middle), and search (right)

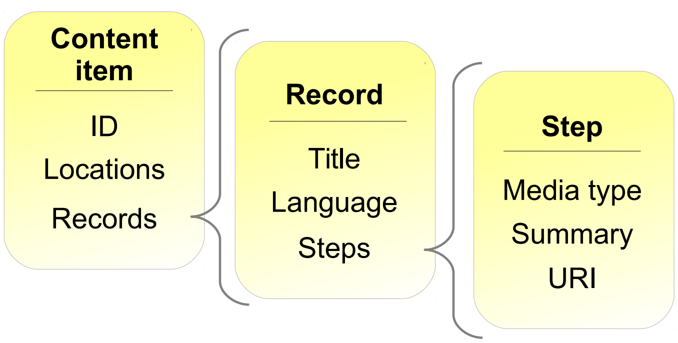


Figure 2. Data model of the content

single Content Item, and so can Records, each representing a particular topic. The topic itself is described by a Record's title together with a language identifier. Language translations of a topic become a new Record. To avoid topics being mixed in the result listing, the results are ordered according to language first, and alphabetically by topic. The user needs analysis recommended further to split content into several steps or segments, and to promote segmented content, something the presented data model is capable of by combining multiple Steps into a Record. A Step has the same language as the Record it belongs to and has one of the Media Types: text, formatted text (HTML), audio, an image or animation, video, or a video with captions. Further elements are a brief Summary and an URI/URL pointing to the media itself. The URI can point to a server that is part of the CMS, but it may also point to external resources (e.g., a video on YouTube). For such external resources, the CMS effectively functions as a service holding metadata on indexed resources. This model supports multiple media types not only for the same Step but also mixing of media types per Record (i.e., several Steps) depending on what type suits a particular step best. For instance, video might be best suited to illustrate a movement, while often a still image is beneficial for highlighting a specific region of a visual.

MobileSage is about just-in-time guidance and *on-demand* assistance. Based on suggestions from the primary-user studies, it was deemed too intrusive to let the mobile application initiate requests for help based on the location of the phone at points of interest, nearby radio fields, etc. Thus, the user indicates a wish for assistance either by scanning a QR code or NFC tag, or by sending a text phrase to the CMS. In the former case, the code or tag carries the ID of a particular Content Item that is read by the mobile app and sent to the CMS, resulting in a list of all topics associated with that ID. Regarding the search phrase, topics are viewed as relevant regardless of the ID, accounting for both Record titles and Step summaries.

One of the challenges of MobileSage is to find *relevant* content and not to confuse the user with extraneous information [32], which helps individuals with orientation and problem solving challenges. The key to this problem is determining the user's context, in terms of location, time and date, user habits, and other aspects. Nearby objects are considered relevant in the CMS by calculating a proximity radius around the user's current position; only Content Items with a location within this circle are returned as results to the phone. The exact radius of the circle was based on heuristics and set to roughly 40 m.

Records are sent in “pages” to the phone, meaning that HoD tells the server how many results per request to return. This is done first of all for practical reasons, i.e., bandwidth limitation, and second because the user is likely to be interested only in the most relevant results, which are presented first. The client on the phone keeps track of the number of transmitted records and is hence able to request a particular page with results, say, page 3 with the records 21 through 30 in case of 10 allowed records per page. If the user scrolls down while being at the bottom of the results list, the client fetches more results if available.

For simplicity, any media content is offered to the HoD as a file for download through HTTP. While this works great with text-based content, the performance in terms of responsiveness of the playback on the media player is suboptimal when connected over a channel with very limited capacity, as discussed in Section VIII-B because media downloads in most clients have to finish before the media is rendered on an output device. Clients that support (true) media streaming and pseudo streaming methods like HTTP Live Streaming will start rendering the output as soon as sufficient data become available. These methods would require the proper setup of a streaming media server.

VII. Content Production

This section focuses on the production of content for MobileSage in particular, and educational and instructive content in general.

The content found should be relevant, concise, and comprehensive. However, as recent research surveys show [33–36], it is extremely difficult to develop methods that can check exactly these properties in a satisfactory manner. MobileSage offers a manual approach in its CMS [32]. As mentioned before, the splitting of longer content into shorter steps is encouraged. The content producer now provides the content abstraction on two levels: A summary of the step itself, and a title wrapping-up of the entire record (see Figure 4). The content producer must tag the content with the proper descriptors for language and its location, if applicable.

Currently, the content must be uploaded in a format accepted by Android OS. This applies to both the format of the content tracks, be it video, audio, or captions, and the format of the embedding media container. In the future, the CMS could be extended to accept any format with transcoding into a proper format supported by the OS.

While the video material could be presented at any resolution, we chose to encode video with a resolution of 480×360 pixels and a bitrate of roughly 200 Kbps @12 fps in H.264 Baseline Profile Level 2.2 format, and embedded in an MP4 transport container. The audio tracks (both stand-alone and as part of a video) had a rate of roughly 48 Kbps @22 KHz mono and were encoded in AAC format. A video containing one visual and one audio track thus had a bitrate of roughly 260 Kbps, which includes overhead data used for track muxing and the container format. The length of the content varied from approximately 10 s to 4 min, but most of the content was between 30 s and 45 s.

We used open-captioning, where the titles (captions) of the voices’ transcript are always visible on the screen, instead of composing a separate captions track. This avoided the extra work of producing a captions file and ensured that the video player always worked properly, without requiring the user to turn it on. We chose a “slab serif” font that was originally designed for fax machines, with a size of at least 36 points. One disadvantage of this approach is that more space is taken up on the content server to store each captioned video, but as the videos were short and at a low resolution, we believe this is a minor issue.

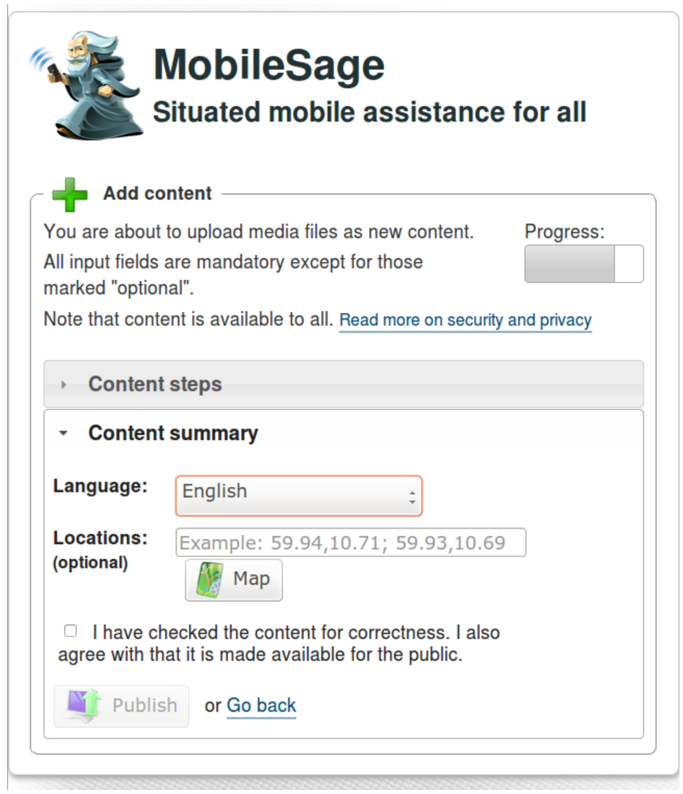


Figure 4. Screenshot from the uploading of new content to the CMS (V1.0)

The content is currently provided in a single quality in terms of resolution, sampling frequency and bitrate as mentioned above. Both have implications for the bandwidth necessary to transmit a given content file to the client’s media player: the larger a picture (in pixels), and the higher the audio sampling frequency (in Hz), the more bandwidth will be necessary. Likewise, the higher the encoder bitrate (in bps), the more bandwidth is required. Channel capacity of the cellular link, however, is a limited resource for physical reasons. It takes time to transmit a particular amount of data over a channel, which also has an impact on the user experience. The service is thus required to respond to user interaction within a reasonably short time span [30].

This requirement is honored by measuring the duration of packet downloads at the client side (which may vary over time according to the signal strength and coverage), and including this information in the server requests, together with information about the phone’s screen size. Content can now be provided in several resolutions or sampling frequencies, and several bitrates. Media searches at the server side can then use this information about the channel conditions and limit the results to media qualities that meet the bandwidth constraints. For example, a phone with a 480×800-pixel screen is connected to the network over a GSM (2G), and the bandwidth averaged over 20 s is measured to be 100 Kbps. Based on the different resolutions and bandwidth, the server decides that a 480×360-pixel video will still render acceptable to good quality. Yet, the 480×360-pixel video comes in two different bitrates; one encoded at a rate of 260 Kbps, and one encoded with 130 Kbps (assuming a constant encoding bitrate). The latter is closest to (but still above) the estimated channel capacity and will be sent to the phone to minimize the service's responsiveness, together with a notification about poor channel conditions.

VIII. User evaluations

The MobileSage prototype was developed in three major iterations where the release of a software deliverable – dubbed Beta, V1.0, and V2.0 – marked the end of an iteration. Each release was evaluated involving end-users. The first and the second evaluations were carried out in Norway, Romania, and Spain; the last evaluation was conducted in Norway only. In total, around 70 informants were part of the evaluations in the three countries.

The subsequent sections summarize the findings from all iterations, with a focus on the set up in Norway.

A. Beta setup

The Beta evaluation in Norway consisted of a travel situation at a subway station in Oslo where participants used the Beta prototype to find help with getting to a subway station, finding a ticket machine, buying and validating a ticket, and choosing the correct platform. We created content for all of these scenarios, with a minimum of audio and video for each. All but one of the scenarios had captioned video, and some had a textual modality in addition. Each of audio, text, and video was done both in Norwegian and in English to allow users to choose an additional content language. We then created several NFC tags for each of the scenarios and used it as a way of getting the information. Testing of QR codes was postponed to a later evaluation due to its unreliability in the Beta version of the app. We tested on two smartphones with an Android OS 4.1 and screen sizes of 480×800 and 720×1184 pixels without any discernible difference in the results.

Eight participants were recruited for the evaluation. They were between 65 and 76 years old, and four of them had no experience with smartphones; however, they were somewhat experienced with computers, and a few of them were familiar with the area. A session consisted of a short introduction to the MobileSage idea, followed by a brief interview concerning their experience with mobile phones. Then, we demonstrated the application and let the informants work on the tasks. One evaluator took notes, while the other would guide the participant to make sure a task wasn't forgotten. After completing all of them, there was a short follow-up interview about the service and the participant's experience about it.

B. Beta results

In the first task, the participants had to create a profile that matched their preferences for text size, language, and types of media they wanted to receive help in. The users understood the concept of several content languages, and the majority (75%) added English to their profile. The user-specific media types ranged from a single one to the entire range as detailed in Section VI, where captioned video, i.e., the richest media type, was chosen most often. The majority (90%) of test persons checked 4–5 media types including audio, even though some participants said they would avoid wearing earbuds or headphones. Text was never requested. Choosing video and captioned video was inconsistent, hinting on a potential misunderstanding of the user concerning the meaning of "captioning". It was recommended to improve the description of media types or show brief examples of them. All participants but one expressed that making the profile was sufficiently easy.

The second task concerned navigation, where the participants had to get from their current location to the nearest subway station. All were able to enter the information needed, but the phone's ability to find the participants' location was unreliable, sometimes placing a participant a block further south or facing the wrong direction. This issue sorted itself out when walking to the location.

The next task dealt with getting help at the ticket machine. Two participants were not able to finish this task due to a technical issue that caused no results to be returned from the CMS, which was corrected subsequently. All others succeeded with using NFC tags or by manually searching for information about where the ticket machine was, how to purchase a ticket, how to validate the ticket, and which platform they had to go to. Though only one was familiar with the technology, two had heard about it, and the rest were unaware of what it was; all participants really liked the technology and experienced it as easy to use.

One problem encountered was the effect the environment had on the signal strength in the phones. While above ground, it was possible to get video and audio without any issues, and the selected item would show up almost instantly. Yet, underground in the subway station, it became very troublesome for the phone to contact the content server. The main reason for this is that the only connections that are currently available in this particular station are so-called Edge (2G) connections, which are much slower compared to a 3G connection, and also very latent. This was no big issue when retrieving, say, the results list. Participants had to wait a long time, though, if they wanted to watch a video. The audio fared a little better, but downloading would not always complete. Sometimes, the application on the phone would simply give up and it would be necessary to download the audio or video from the beginning. Most participants noted that it took a while to get the information in this case. With the continuing widespread of 2G connections in many countries, it is recommended to produce at least one version of low-resolution low-bitrate content, and to use techniques that increase the responsiveness of media players, such as media streaming, as discussed in Section VI.

No users complained about the size, resolution, quality, frame rate, or length of the video. Some participants noted that the font used for the captions indeed was sufficiently large and easy to read. There was only one instance where people commented on unclear information, where a video showed an unreadable display on a ticket machine.

All participants felt that a help-on-demand system was something that would be useful for them. One even claimed that she was scared of using the ticket machine and always went to a counter instead, but now she would continue to use the machine since she felt confident to manage buying a ticket based on the app and the provided instructions. Concerning potential improvements, the most popular suggestions were a shorter response time for videos (when in the subway station) and dynamic information, such as time schedules. Those familiar with mobile applications suggested to include MobileSage's functionality in the public transport provider's current smartphone application.

C. V1.0 setup

In the V1.0 evaluation all of the attention was given to the user experience when using the HoD app. The Norwegian evaluation involved 10 informants from the local senior user group, aged 67 to 83, from both genders, all with varying ICT and mobile-phone experience (though none were novice ICT users), and a few with a mother tongue different from Norwegian. All informants received a small financial gratituity for the participation. Two Android phones (Galaxy Nexus and Nexus S) were used in the tests.

The scenario's focus was on matters not tested in the last trials and emphasized multilingual content, the concept of steps, and QR codes. The following tourist situation was considered: A visitor to Norway arrives at the tourist information center in Oslo, and a poster mentioning the famous Kon-Tiki Museum catches the visitor's eye. The poster provides both an NFC tag and a QR code. The visitor scans either of them with the MobileSage app, and several pieces of information are presented: information about the museum and how to get there from the user's current position, how to buy a ticket at the nearby machine, how to find the proper bus stop, when the next bus is arriving, and when to get off the bus while riding to the museum.

Most of these five steps were presented in multiple modalities, such video, audio, and formatted text, others were available just in a single media type. The latter two steps – the expected duration of the waiting time for the next bus and the expected duration of the arrival of the bus at the proper bus stop – showed dynamic content (real-time data) from the servers of the municipality's transport company Ruter. This was achieved by HTML redirects from the content provided by the CMS to Ruter's server.

The informants were first briefly introduced to the MobileSage idea in general, and scanning of NFC/QR in particular. After that we went to the nearby tourist information where we had hung up some of the poster as described above, and simply watched as the participants went through the steps of travelling to the museum. In case of problems we would also assist the user with clarification and also give some technical aid (Figure 5). Throughout a single trial, the participant moved from the poster to the ticket machine, and then to the bus stop; the last step (the bus ride to the museum) was simulated only for practical reasons. After that, the user was questioned about their user experience and had to fill out a brief questionnaire to gather their opinion.

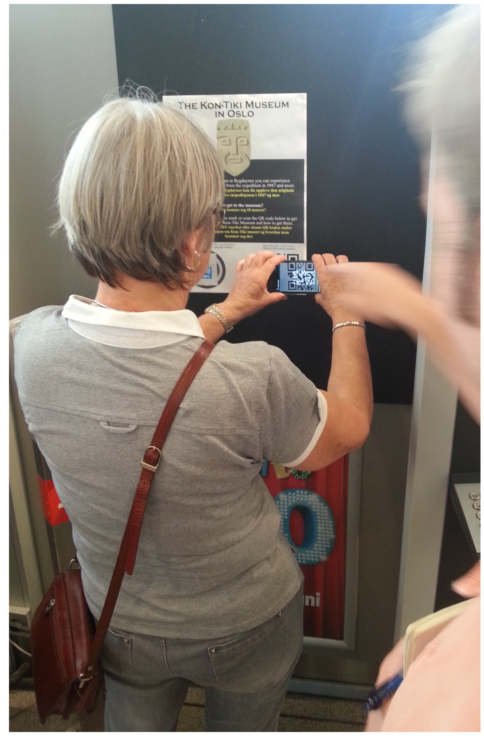


Figure 5. Example situation in a user trial, a participant scans a QR code

D. V1.0 results

Subsequently, the findings from the Norwegian trials are presented as an excerpt from a larger report [37].

Overall, the English MobileSage version was acceptable for the English-speaking testers, even though they commented on several non-translated page elements in the dynamic webpages from the travel company. Integration of services, including the proper communication of the user's language, is key here, besides the mandatory translation of all language strings.

The participants found the prototype in general accessible, but there were several issues related to real-life situations: traffic, crowd, and town noise was a problem when trying to hear the sound of the videos, both indoors and outdoors. All participants would use the relatively weak speakers in the smartphones. As one of the participant remarked, "elderly never use headphones, you know." Here, video captions were useful to the participants. Next, bright outside sunlight reduced the screen contrast, making it difficult to read what was displayed. Here, automatic adjustment of the display's brightness and contrast and improved displays would help, but this is beyond the scope of this project. Some participants found the text size and also the virtual-keyboard letters too small. However, even though it had been pointed out to participants that they could adjust the settings according to their own preferences, none actually changed the default text size. The size of the keyboard letters could not be changed, and this might be the reason why the users found NFC and QR codes so attractive when searching for information.

The informants all agreed that the ability to customize and personalize media modalities and output was valuable for them and other elderly as well. Due to time constraints, though, this topic was not tested systematically. The fact that no user changed the settings shows on the other hand the necessity for suitable default values, such as captioned videos as default media type as it combines a visual with audio and text.

Regarding adaptivity, the trial observers noticed that the most used functionality was automatically placed in a prominent position in the user interface (on top). However, the participants did not seem to notice. We did collect usage data for each participant, but due to the small duration of each trial, a sufficient amount of data for each participant was never generated. Future work should test out adaptivity in a realistic manner.

The participants had quick access to the content and were all satisfied with the response time. It surely helped that the evaluations were held in an area with a good GSM signal, but contributing to this was also the strategy to switch from plain downloading in the first trial to HTTP pseudo streaming, which shortens the time after which the media player starts playing drastically.

Most informants had heard about QR codes or at least said they had noticed them, but very few knew about NFC, let alone its logo. All participants preferred scanning over text-based search during the trial. Here, nine out of 10 found that NFC was easier to use than QR due to a shorter response time. With QR, many found it cumbersome to find the correct distance and angle between the smartphone camera and the QR code on the poster. In contrast to the beta trial, NFC tag scanning went smoother, mainly because we now carefully had placed the tags with some distance to any metal surfaces.

As opposed to the beta evaluation, the app now rendered the content of the result directly if only one had been found. Most participants preferred this, but were in turn confused when the result consisted of several steps, as showing a step overview had been omitted. Other than that, steps as a concept was well understood and accepted.

As one of the outcomes from the user interviews, Table I shows the general user acceptance, measured by means of the System Usability Scale (SUS). The table clearly documents the improvements in user experience on almost all topics as compared to the Beta version. The largest positive change occurs related to the app's ease of use, with an increase from 3.1 (uncertain) to 4.4 (clear agree) average score. Overall, our participants had a positive view on the MobileSage system and found it useful and relevant. The scale also shows potential for improvement, however, when it comes to opinions concerning the app's ease of use.

E. V2.0 setup

The following paragraphs are an excerpt from the more detailed evaluation report [38].

In the V2.0 evaluation, we looked at the final version of the CMS to see how well it worked, and the final version of the HoD app to get feedback on changes to the app. It was a limited evaluation held in cooperation with three participants from the local senior user group that had been involved in the project. Due to time constraints, the evaluation was held as a workshop with all three participants using the system and giving comments simultaneously.

The participants were given a short introduction about the CMS and were then asked to create instructional content that they subsequently uploaded to the CMS. Participants chose to create a video in three parts (steps) for heating water in a microwave. They shot the video on one of the phones, copied the videos over to laptop, and proceeded to put them on the CMS.

For the HoD part, many of the tasks were borrowed from the previous evaluation. We gave the informants a short introduction to the HoD, specifically focusing on the media types setting. The participants were then instructed to find a video about tickets by using the scanning functionality after having changed the media type settings to accept only formatted text and plain text.

The session was concluded with a short discussion about the app and the CMS.

F. V2.0 results

Concerning the CMS, participants were in the beginning confused about how content was organized and had in particular problems to understand that a record can consist of multiple steps. Related to this is that a user needs to fill in two titles, one for a step, and another for the entire record. This information was not included on the web page itself and needed to be explained by the trial observers. Once the concept was understood, participants were able to create and add content containing multiple steps.

Participants encountered problems related to the media type of the content to upload. Depending on the file type (MIME type), particular buttons ("Continue", "Publish") were shown or hidden. As mentioned in Section VII, only a limited number of formats are allowed to be uploaded in the CMS. However, the negative result from the media type check was not communicated to the user who then had to assume that the upload form did not work. Hence, the user needs to be properly informed about each requirement to an input field, and each conducted check. On the other hand, the participants were able to complete all the steps and fill in the content summary, but they could not actually add the content to the CMS.

All participants commented on the necessity of extra information for creating and adding content in terms of tips on how to create videos so the videos would be most instructional. One participant was afraid to pack too much information into the content. A take-away here may be to provide thought-through tools and also assistance for proper content generation.

The CMS makes content available for all, which turned out to be of concern for some participants. They commented that users could hesitate to upload information if it was public for everyone, even though the trial observers pointed out that the location information helps to limit exposure for information, and that the majority of all information intended for MobileSage is designed to be for public access. It is noted here that it is technically possible to restrict access to a record say with a passphrase, but this of course adds to the complexity of the system.

TABLE I. THE SYSTEM USABILITY SCALE COMPARISON BETWEEN BETA AND V1.0; 1=STRONGLY DISAGREE, 5=STRONGLY AGREE.

Question	Beta	V1.0
I think that I would like to use this system frequently	3.5	4.0
I found the system unnecessarily complex	1.6	1.7
I thought the system was easy to use	3.1	4.4
I think that I would need the support of a technical person to be able to use this system	1.7	1.9
I found the various functions in this system were well integrated	3.0	3.8
I thought there was too much inconsistency in this system	2.5	1.4
I would imagine that most people would learn to use this system very quickly	3.8	4.3
I found the system very cumbersome to use	1.3	1.3
I felt very confident using the system	2.6	3.2
I needed to learn a lot of things before I could get going with this system	3.5	2.3

In the settings of the app, when choosing media types, users were presented with the jargon terms for the types in the database and not a suitable translation in their own language, leading to questions about the meaning of each term. The built-in help in the app does indeed explain these types without using jargon, but this help is not available when choosing types. The conclusion here is that technical jargon should be avoided, and that concise and explaining help should be available where challenges might arise.

Using the different help-on-demand functions in the HoD app, including scanning of QR codes and NFC tags, worked as expected. One phone, however, was a bit large, and users had to move the phone forth and back to get to read the tag. We believe this problem would vanish as the users become used to their phone.

Generally, the participants were excited about the possibilities of the MobileSage service and wanted to use both systems (app and CMS) more. We are going to address a number of the issues found in the various evaluations before a final version of each system is released. The CMS is currently available online [39], and the HoD app will shortly be offered through Google Play [40] and the MobileSage website [41].

IX. CONCLUSION AND OUTLOOK

We presented MobileSage, a service for delivery of context-aware, accessible, and personalized help content in an on-demand manner, examplified by two prototypes, a smartphone application and a content management system.

In the prototypes, we incorporated results from related research as well as the needs of the primary, secondary, and tertiary users. The system employs multimodality as an accessibility measure, as well as internationalization and data mining for user personalization, multi-resolution and multi-rate transmission techniques for device adaptivity, and location-aware media searches for relevance. The system can index both internal and external media databases.

The findings from our focus groups and the three user evaluations show that there is a strong desire for context-dependent and adaptive help content, and that such a service is highly appreciated by its end-users. Moreover, the high degree of user involvement in the project ensured products which reflect the wishes and needs of all stakeholders in general, and the primary users in particular. The prototypes have improved quite a bit throughout the user evaluations, as reflected by increases in the System Usability Scale.

At the end of the project, we now have the proof of concepts and the proper technology with development maturity, as according to the AAL JP vision. A final service with production maturity could be released within 1–2 years time. Among remaining issues are: the proper handling of offline and low-signal situations, proper tools and an improved user interface for the generation of content, including content access control, and assistance and guidelines for the production of usable and accessible content, such as the length of timed media, phrasing of information, and how multimedia content should be organized. Also, the integrity of the content is currently not controlled in any way. Here, a review system and content credibility management could offer a proper solution.

Another area for future exploration is a long-term evaluation of the adaptation module. All of our evaluations were too short to see how much adaptivity affected the participants. One way to solve this is to provide the HoD app to users for a longer period (weeks or months), to see how the app adapts to their usage, and if users would notice and appreciate these changes.

ACKNOWLEDGMENTS

This work was partly funded by the European Commission through the AAL Joint Programme, the Norwegian Research Council, and national bodies in Spain and Romania. The authors thank the consortium members for their valuable contributions and all individuals involved in the user studies for their feedback.

REFERENCES

[1] T. Halbach and T. Schulz, "Mobilesage – A prototype based case study for delivering context-aware, personalized, on-demand help content," in *Proceedings of the Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services*, IARIA. Venice (Italy): IARIA XPS Press, Oct. 2013. [Online]. Available: http://www.iaria.org/conferences2013/CENTRIC13.html

[2] D. Metz and M. Underwood, *Older, Richer, Fitter: Identifying the Consumer Needs of Britain's Ageing Population*. Age Concern England, 2005.

[3] European Commission, "Digital Agenda: Commission proposes rules to make government websites accessible for all," retrieved 2014-05-20. [Online]. Available: http://europa.eu/rapid/press-release_IP-12-1305_en.htm

[4] AAL Association, "Innovative ict solutions for ageing – ambient assisted living," retrieved 2014-05-20. [Online]. Available: http://www.aal-europe.eu/

[5] ——, "Ict-based solutions for advancement of older persons' independence and participation in the "self-serve society"," 3rd Call for Proposals (2010). [Online]. Available: http://www.aal-europe.eu/

[6] T. H. Røssvoll, "The European MobileSage Project – – Situated adaptive guidance for the mobile elderly," in *Electronic Government and Electronic Participation, Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart 2012*, H. J. Scholl, L. S. Flak, M. Janssen, A. Macintosh, C. E. Moe, Øystein Sæbø, E. Tambouris, and M. A. Wimmer, Eds., vol. Schriftenreihe Informatik 39, International Federation for Information Processing. Kristiansand (Norway): Trauner Verlag, Sep. 2012, pp. 215–222.

[7] APSIS4ALL Project Consortium, "The APSIS4ALL project," retrieved 2014-05-20. [Online]. Available: http://www.apsis4all.eu/

[8] ASK-IT Project Consortium, "The ASK-IT project," retrieved 2014-05-20. [Online]. Available: http://www.ask-it.org/

[9] MyUI Project Consortium, "The MyUI project," retrieved 2014-05-20. [Online]. Available: http://www.myui.eu/

[10] Guide Project Consortium, "The Guide project," retrieved 2014-05-20. [Online]. Available: http://www.guide-project.eu/

[11] DIADEM Project Consortium, "The DIADEM project," retrieved 2014-05-20. [Online]. Available: http://www.project-diadem.eu/

[12] GPII Project Consortium, "The GPII project," retrieved 2014-05-20. [Online]. Available: http://gpii.org/

[13] Snapi Project Consortium, "The Snapi project," retrieved 2014-05-20. [Online]. Available: http://www.snapi.org.uk/

[14] CEN (European Committee for Standardization) TC224 WG6, "En-1332-4: Coding of user requirements for people with special needs," 2012.

[15] GoldUI Project Consortium, "The GoldUI project," retrieved 2014-05-20. [Online]. Available: http://www.goldui.eu/

[16] WayFiS Project Consortium, "The WayFiS project," retrieved 2014-05-20. [Online]. Available: http://www.wayfis.eu/

[17] Mediate Project Consortium, "The Mediate project," retrieved 2014-05-20. [Online]. Available: http://www.mediate-project.eu/

[18] Access2all Project Consortium, "The Access2all project," retrieved 2014-05-20. [Online]. Available: http://www.access-to-all.eu/

[19] A. J., S. J.L., and A. J.I., "Ambiennet – ambient intelligence supporting navigation for people with disabilities," *Jornada de Seguimiento de Proyectos*, 2009.

[20] HAPTIMAP Project Consortium, "The HAPTIMAP project," retrieved 2014-05-20. [Online]. Available: http://www.haptimap.org/

[21] MAPPED Project Consortium, "The MAPPED project," retrieved 2014-05-20. [Online]. Available: http://services.txt.it/MAPPED

[22] E. M., I. M., and L. I., "Touch- and audio-based medication management service concept for vision impaired older people," *IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, 2011.

[23] V. G., K. D., S. O., M. S., and T. S., "Indoor localization using passive RFID," *Proceedings of Signal Processing, Sensor Fusion, and Target Recognition*, 2011.

[24] K. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, and I. Solheim, "A user profile ontology based approach for assisting people with dementia in mobile environments," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 6390–6393.

[25] W. Burns, L. Chen, C. Nugent, M. Donnelly, K.-L. Skillen, and I. Solheim, "A conceptual framework for supporting adaptive personalized help-on-demand services," in *Ambient Intelligence*. Springer, 2012, pp. 427–432.

[26] K.-L. Skillen, L. Chen, C. Nugent, M. Donnelly, W. Burns, and I. Solheim, "Using swrl and ontological reasoning for the personalization of context-aware assistive services," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2013, p. 48.

[27] W. Burns, L. Chen, C. Nugent, M. Donnelly, K. L. Skillen, and I. Solheim, "Mining usage data for adaptive personalisation of smartphone based help-on-demand services," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2013, p. 39.

[28] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, "Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments," *Future Generation Computer Systems*, 2013.

[29] Øystein Dale, "User needs analysis," MobileSage Consortium, Tech. Rep. MobileSage Deliverable D2.1, 2012. [Online]. Available: http://mobilesage.eu/public-documents/public-deliverables

[30] T. H. Røssvoll, "User requirements specification," MobileSage Deliverable D2.2, MobileSage Consortium, Tech. Rep., Feb. 2012. [Online]. Available: http://mobilesage.eu

[31] L. Curescu, I. Anghelache, and T. H. Røssvoll, "System requirements specification for help-on-demand service," MobileSage Deliverable D2.3, MobileSage Consortium, Tech. Rep., Apr. 2012. [Online]. Available: http://mobilesage.eu

[32] T. H. Røssvoll and V. A. Gracia, "System requirements specification for content management service," MobileSage Deliverable D2.4, MobileSage Consortium, Tech. Rep., Apr. 2012. [Online]. Available: http://mobilesage.eu

[33] R. Mohamad Rasli, S. C. Haw, and R. Mohamad Rasli, "A survey on optimizing image, video, and audio query retrieval in multimedia databases," *International Journal of Advanced Computer Science*, vol. 2, no. 6, 2012.

[34] D. Das and A. F. Martins, "A survey on automatic text summarization," *Literature Survey for the Language and Statistics II course at CMU*, vol. 4, pp. 192–195, 2007.

[35] D. Brezeale and D. J. Cook, "Automatic video classification: A survey of the literature," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 3, pp. 416–430, 2008.

[36] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 6, pp. 797–819, 2011.

[37] I. Solheim, T. Halbach, T. Schulz, J. R. Simon, I. Turcu, A. Sterea, I. Anghelache, and L. Spiru, "D4.3 evaluation report," MobileSage Deliverable D4.3, MobileSage Consortium, Tech. Rep., Oct. 2013. [Online]. Available: http://mobilesage.eu

[38] T. Schulz, "CMS & HOD test at Seniornett," MobileSage Consortium, Tech. Rep. Internal MobileSage Report, 2014.

[39] MobileSage Consortium, "Content management system." [Online]. Available: http://mobilesage.nr.no

[40] ——, "Help-on-demand application," available after 2014-04-01. [Online]. Available: https://play.google.com/store/search?q=mobilesage

[41] V. Sánchez, "The MobileSage Project," retrieved 2014-05-20. [Online]. Available: http://mobilesage.eu

International Journal on Advances in Intelligent Systems, vol 7 no 1 & 2, year 2014, http://www.iariajournals.org/intelligent_systems/

Trustworthy Autonomic Architecture (TAArch): Implementation and Empirical Investigation

Thaddeus Eze and Richard Anthony

Autonomic Computing Research Group
School of Computing & Mathematical Sciences (CMS)
University of Greenwich, London, United Kingdom
{T.O.Eze and R.J.Anthony}@gre.ac.uk

***Abstract* — This paper presents a new architecture for trustworthy autonomic systems. This trustworthy autonomic architecture is different from the traditional autonomic computing architecture and includes mechanisms and instrumentation to explicitly support run-time self-validation and trustworthiness. The state of practice does not lend itself robustly enough to support trustworthiness and system dependability. For example, despite validating system's decisions within a logical boundary set for the system, there's the possibility of overall erratic behaviour or inconsistency in the system emerging for example, at a different logical level or on a different time scale. So a more thorough and holistic approach, with a higher level of check, is required to convincingly address the dependability and trustworthy concerns. Validation alone does not always guarantee trustworthiness as each individual decision could be correct (validated) but overall system may not be consistent and thus not dependable. A robust approach requires that validation and trustworthiness are designed in and integral at the architectural level, and not treated as add-ons as they cannot be reliably retro-fitted to systems. This paper analyses the current state of practice in autonomic architecture, presents a different architectural approach for trustworthy autonomic systems, and uses a datacentre scenario as the basis for empirical analysis of behaviour and performance. Results show that the proposed trustworthy autonomic architecture has significant performance improvement over existing architectures and can be relied upon to operate (or manage) almost all level of datacentre scale and complexity.**

Keywords - trustworthy architecture; trustability; validation; datacentre; autonomic system; dependability; stability; autonomic architecture

I. INTRODUCTION

A robust autonomic architecture is a vital key to achieving dependable (or trustworthy) autonomic systems. We have made initial progress [1] in this direction to address the issue of autonomic trustworthiness through adequate run-time conformance testing as integral part of a trustworthy autonomic architecture (different from the traditional autonomic architecture). This work is an extension of the initial progress and the implementation (with empirical analysis) of the new trustworthy architecture. The traditional autonomic architecture as originally presented in the autonomic computing blueprint [2] has been widely accepted and deployed across an ever-widening spectrum of autonomic system (AS) design and implementations. Research results in the autonomic research community are based, predominantly, on the architecture's basic MAPE (monitor-analyse-plan-execute) control loop, e.g., [3][4]. Several implementation variations of this control loop, for example [5][6], have been promoted. While [5] breaks the MAPE components into two main groups with the Monitor/Analyze group handling reactive tasks and the Plan/Execute group responsible for proactive adaptation, [6] adopts a slightly different approach. In [6], the MAPE architecture is divided into global and local sub-architectures, which implement Analyze/Planning and Monitor/Execute components, respectively. Alternative approaches, e.g., the intelligent machine design (IMD) based approach [7] have also been proposed. However, research [8] shows that most approaches are MAPE [9] based. Despite progress made, the traditional autonomic architecture and its variations is not sophisticated enough to produce trustworthy ASs. A new approach with inbuilt mechanisms and instrumentation to support trustworthiness is required.

At the core of system trustworthiness is validation and this has to satisfy run-time requirements. In large systems with very wide behavioural space and many dimensions of freedom, it is close to impossible to comprehensively predict possible outcomes at design time. So it becomes highly complex to make sure or determine whether the autonomic manager's (AM's) decision(s) are in the overall interest and good of the system. There is a vital need, then, to dynamically validate the run-time decisions of the AM to avoid the system '*shooting itself in the foot*' through control brevity, i.e., either too loose or too tight control leading to unresponsive or unstable system respectively. The traditional autonomic architecture does not explicitly and integrally support run-time self-validation; a common practice is to treat validation and other needed capabilities as add-ons. Identifying such challenges, the traditional architecture has been extended (e.g., in [10]) to accommodate validation by integrating a *self-test* activity into the autonomic architecture. But the question is whether validation alone can guarantee trustworthiness.

The need for trustworthiness in the face of the peculiar nature of ASs, (e.g., context dynamism) comes with unique and complex challenges validation alone cannot sufficiently address. Take for instance; if a manager (AM) erratically changes its decision, it ends up introducing noise to the system rather than smoothly steering the system. In that instance, a typical validation check will *pass* each correct decision (following a particular logic or rule) but this could lead to oscillation in the system resulting in instability and inconsistent output, which could emerge at a different logical level or time scale. A typical example could be an AM that follows a set of rules to decide when to move a server to or from a pool of servers; as long as the conditions of the rules are met, the AM will move servers around not minding the frequency of changes in the conditions. An erratic change of decision (high rate of moving servers around) will cause undesirable oscillations that ultimately detriment the system. What is required is a kind of intelligence that enables the manager to smartly carry out a change only when it is safe and efficient to do so – within a particular (defined) safety margin. A higher level of self-monitoring to achieve, for example, stability over longer time frames, is absent in the MAPE-orientated architectures. This is why autonomic systems need a different approach. The ultimate goal of the new approach is not just to achieve self-management but also to achieve consistency and reliability of results through self-management. These are the core values of the proposed architecture in this paper.

We look at the background of work towards AS trustworthy architecture in Section II. We present a new trustworthy autonomic architecture in Section III and present a datacentre-based implementation and empirical analysis of the new architecture in Section IV. Section V concludes the work.

II. BACKGROUND

The idea espoused in this work is that trustworthiness (and any other desired autonomic capability) should be conceived at design stage. This means that the autonomic architecture should be flexible (and yet robust) enough to provide instrumentations that allow designers to specify processes to achieve desired goals. It then follows that we need to rethink the autonomic architecture. In this section, we look at the current state of practice and efforts directed towards AS trustworthiness. We analyse few proposed trustworthy architectures and some isolated bits of work that could contribute to trustworthy autonomic computing. Trustworthiness requires a holistic approach, i.e., a long-term focus as against the near-term needs that merely address methods for building trust into existing systems. This means that trustworthiness needs to be designed into systems as integral properties.

Chan *et al.* [11] asks the critical question of "How can we trust an autonomic system to make the best decision?" and proposes a '*trust*' architecture to win the trust of autonomic system users. The proposal is to introduce trust into the system by assigning an "instantaneous trust index" (ITI) to each execution of a system's AM –where ITI could be computed, for example, by examining what fraction of AM suggested actions the user accepts unchanged, or by examining how extensive the changes that the user makes to the suggested actions are. The overall trust index, which reflects the system administration's level of trust in the AM, is computed as the function f(ITIi) where $i = 1, 2, 3, \ldots$ and ITIi are the individual ITIs for each AM execution. This is similar to the proposal in this work in the sense that it considers trust as architecture-based and also defines trust in the language of the user. However, this method will be overly complex (and may be out of control) in large systems with multiple AMs if the user is required to moderate every single AM suggested action. In such systems some of the AM's decisions are not transparent to the human user. Another effort that supports the idea that dependability should be conceived at design time and not retro-fitted to systems is the work in [12]. Hall and Rapanotti [12] propose an *Assurance-Driven Design* and posit that engineering design should include the detailing of a design for a solution that guarantees satisfaction of set requirements and the construction of arguments to assure users that the solution will provide the needed functionality and qualities. The key point here is that trustworthiness is all about securing the confidence of the user (that the system will do what it says) and the way to achieve this is by getting the design (architecture) right. This is the thrust of this work.

Shuaib *et al.* [7] propose a framework that will allow for proper certification of A-C systems. Central to this framework is an alternative autonomic architecture based on Intelligent Machine Design (IMD), which draws from the human autonomic nervous system.

Kikuchi *et al.* [13] proposes a policy verification and validation framework that is based on model checking to verify the validity of administrator's specified policies in a policy-based system. Because a known performing policy may lead to erroneous behaviour if the system (in any aspect) is changed slightly, the framework is based on checking the consistency of the policy and the system's defined model or characteristics. This is another important aspect of the proposed solution in this work –validation is done with reference to the system's defined goal.

A trustworthy autonomic grid computing architecture is presented in [14]. This is to be enabled through a proposed fifth self-* functionality, self-regulating: Self-regulating capability is able to derive policies from high-level policies and requirements at run-time to regulate self-managing behaviours. One concern here is that proposing a fifth autonomic functionality to regulate the self-CHOP functionalities as a solution to AS trustworthiness assumes that trustworthiness can be achieved when all four functionalities perform 'optimally'. This assumption is not entirely correct. The self-CHOP functionalities alone do not ensure trustworthiness in ASs. Take for example; the self-CHOP functionalities do not address *validation*, which is a key factor in AS trustworthiness. The self-CHOP (or sometimes referred to as self-*) stands for self-Configuring, self-Healing, self-Optimising, and self-Protecting. These are the characteristics or functional areas that define the capabilities of autonomic systems and will be referred to as autonomic functionalities in this paper.

Another idea is that trustworthiness is achieved when a system is able to provide accounts of its behaviour to the extent that the user can understand and trust. But these accounts must, amongst other things, satisfy three requirements: provide a representation of the policy guiding the accounting, some mechanism for validation and accounting for system's behaviour in response to user demands [15]. The system's actions are transparent to the user and also allow the user (if required) the privilege of authorising or not authorising a particular process. This is a positive step (at least it provides the user a level of confidence and trust) but also important is a mechanism that ensures that any 'authorised' process does not lead to unreliable or misleading results. This is one aspect not considered by many research efforts. There are possibilities of erratic behaviour (which is not healthy to the system) despite the AM's decisions being approved. One powerful way of addressing this challenge is by implementing a *dead-zone* (DZ) logic originally presented in [16]. A DZ, which is a simple mechanism to prevent unnecessary, inefficient and ineffective control brevity when the system is sufficiently close to its target value, is implemented in [16] using Tolerance-Range-Check (TRC) object. The TRC object encapsulates DZ logic and a three-way decision fork that flags which action (left, null or right) to take depending on the rules specified. The size of the DZ can be dynamically adjusted to suit changes in environmental volatility. A key use of this technique is to reduce oscillation and ensure stability in the face of high rate of adaptability despite process correctness. A mechanism to automatically monitor the stability of an autonomic component, in terms of the rate the component changes its decision (for example when close to a threshold tipping point), was presented in [17]. The *DecisionChangeInterval* property is implemented in the AGILE policy language [17] on decision making objects such as rules and utility functions. This allows the system to monitor itself and take action if it detects instability at a higher level than the actual decision making activity. This technique is used in the proposed solution herein.

Heo and Abdelzaher [18] present 'AdaptGuard', a software designed to guard adaptive systems from instability resulting from system disruptions. The software is able to infer and detect instability and then intervenes (to restore the system) without actually understanding the root cause of the problem –*root-cause-agnostic* recovery. Instability is another aspect addressed in the solution proposed in our work. Because AM control brevity could lead to instability despite process correctness, it is important to also consider this scenario. Hawthorne *et al.* [19] demonstrates Teleo-Reactive (T-R) programming approach to autonomic software systems and shows how T-R technique can be used to detect validation issues at design time and thus reducing the cost of validation issues.

Validation is central to achieving trustworthy autonomics and this has to meet run-time requirements. A generic self-test approach is presented in [10]. The authours of [10] extended the MAPE control loop to include a new function called *Test* (Figure 1). By this they define a new control loop comprising Monitor, Analyse, Decision, Test and Execute –MADTE activities. The MADTE loop works like the MAPE loop only that the Decision activity calls the Test activity to validate a chosen action should it determine to adapt a suggested behaviour. The Test activity carries out a test on the action and returns its result to the Decision activity, which then decides whether to implement, skip or choose another action. (An adaptation is favoured if *Test* indicates that it will lead to component's better performance in terms of characteristics such as optimisation, robustness or security.) The process is repeated if the latter is the case. When an action is decided on, the decision activity passes it to the Execute activity for implementation. This is vital to run-time self-validation and is consistent with our proposed solution in this work in terms of designing validation into the system's architecture. A feedback-based validation, which relies on a kind of secondary (mostly external) expertise feedback to validate the output of a system is presented in [20]. This is reactionary and has no contribution to the result of the system in the first place. Though this may suffice for some specific system's needs, what is generally required for AS validation is run-time validation of decisions (or processes) that lead to system outputs.

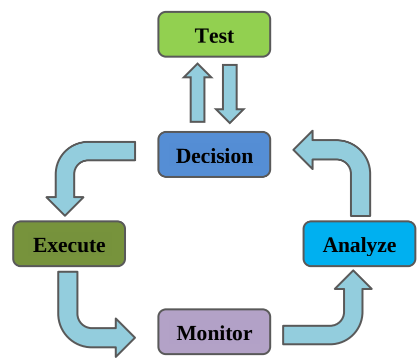


Figure 1: Control loop with a test function [10]

It should be noted that AS trustworthiness goes beyond secure computing. It is result orientated; not focusing on how a goal is achieved but the dependability of the output achieved. All systems, no matter how simple or complex, are designed to meet a need, but not all systems have security concerns. So trustworthiness is not all about security and validation. On the other hand, it is not about showing that a system or process works but also making sure that it does exactly what it is meant to do. This aspect is addressed in the proposed trustworthy autonomic architecture by a component that carries out a longer term assessment of the system's actions. These are the evolving challenges and where work must be concentrated if we are to achieve certifiable autonomic systems.

A. *Autonomic architecture life-cycle*

We argue that trustworthiness cannot be reliably retrofitted into systems but must be designed into system architectures. We track autonomic architecture (leading to trustworthiness) pictorially in a number of progressive stages addressing it in an increasing level of detail and sophistication. Figure 2 provides a key to the symbols used.

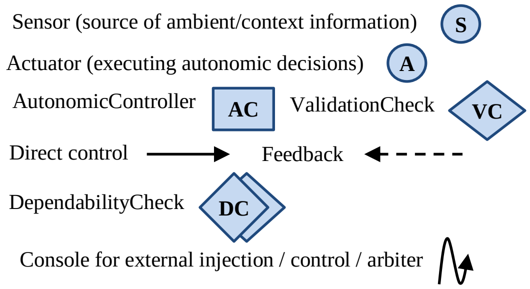


Figure 2: Pictographic key used for the architecture life-cycle.

Figure 3 illustrates the progression, in sophistication, of autonomic architectures and how close they have come to achieving trustworthiness. Although this may not be exhaustive as several variations and hybrids of the combinations may exist, it represents a series of discrete progressions in current approaches.

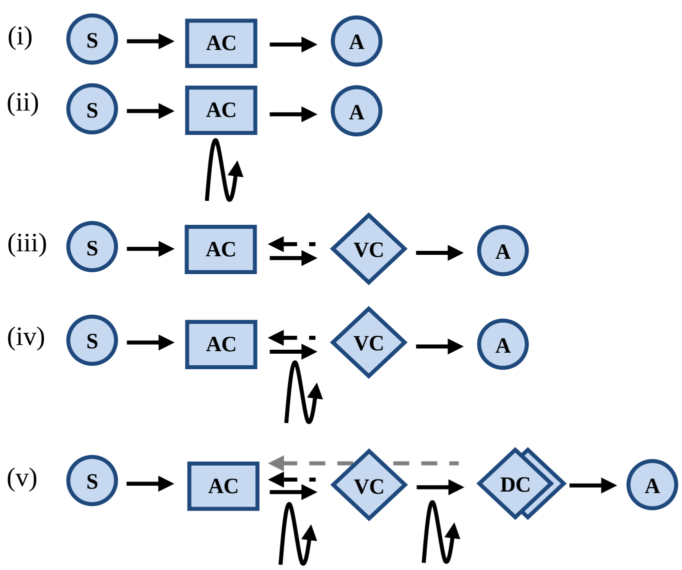


Figure 3: Pictorial representation of autonomic architecture life-cycles.

Two distinct levels of sophistication are identified: The first level represents the traditional autonomic architecture (Figure 3 (i) and (ii)) basically concerned with direct self-management of controlled/monitored system following some basic sense-manage-actuate logic defined in AC. For the prevailing context, AC is just a container of autonomic control logic, which could be based on MAPE or any other autonomic control logic. The original autonomic architecture proposed with the introduction of autonomic computing [2] falls within this level. This achieves basic self-management capability and has since been adapted by several researchers to offer more smartness and sophistication. To add a degree of trust and safeguard, an external interface for user control input is introduced in (ii). This chronicles such approaches that provide a console for external administrative interactions (e.g., real-time monitoring, tweaking, feedback, knowledgebase source, trust input, etc.) with the autonomic process. An example of level (ii) is work in [15], where the system's actions are transparent to the user and the user can moderate the behaviour of the system by allowing or disallowing system decided actions. The system has a console that offers the user the privilege of authorising or not authorising a particular process. Another example in this category is unmanned vehicles (UVs). In UVs there are provisions for activating auto piloting and manual piloting. The user can decide when to activate either or run a hybrid.

The second level (Figure 3 (iii) and (iv)) represents efforts towards addressing run-time validation. Instrumentations to enable systems check the conformity of management decisions are added. This includes such approaches that are capable of run-time self-validation of autonomic management decisions. The validation check is done by the VC component and the check results in either a *pass* (in which case the validated decision is actuated) or a *fail*. Where the check fails VC sends feedback to AC with notification of failure (e.g., policy violation) and new decision is generated. An additional layer of sophistication is introduced in Figure 3 (iv) with external touch-point for higher level of manageability control. This can be in the form of an outer control loop monitoring over a longer time frame an inner (shorter time frame) control loop. The work in [10] (explained in Section II), which is an extension of MAPE control to include a '*Test*' activity corresponds to level (iii) of Figure 3. The *Test* activity tests every suggested action (decision) by the plan activity. If the test fails the action is dropped and a new one is decided again. The work in [21] corresponds to level (iv) of Figure 3. The work in [10] is extended in [21] to include auxiliary test services components that facilitate manual test management and a detailed description of interactions between test managers and other components. Here test managers implement closed control loops on autonomic managers (such as autonomic managers implement on managed resources) to validate change requests generated by the autonomic managers.

At the level of current sophistication (state-of-the-art), there are techniques to provide run-time validation check (for behavioural and structural conformity), additional console for higher level (external) control, etc. Emerging and needed capabilities include techniques for managing oscillatory behaviour in autonomic systems. These are mainly implemented in isolation. What is required is a holistic framework that collates all these capabilities into a single autonomic unit. Policy autonomics is one of the most used autonomic solutions. Autonomic managers (AMs) follow rules to decide on actions. As long as policies are validated against set rules the AM adapts its behaviour accordingly. This may mean changing between states. And when the change becomes rapid (despite meeting validation requirements) it is capable of introducing oscillation, vibration and erratic behaviour (all in form of noise) into the system. This is more noticeable in highly sensitive systems. So a trustworthy autonomic architecture needs to provide a way of addressing these issues. Level (v) of Figure 3 falls within the next level of sophistication required to address the identified issues and ensure dependability. This is at the core of the proposed solution presented in next the Section.

III. TRUSTWORTHY AUTONOMIC ARCHITECTURE

This section presents the new trustworthy autonomic architecture (TAArch). First, a general view of the architecture is presented and then followed by detailed explanation of its components. Figure 4 explains a trustworthy autonomic framework with three major components that embody self-management, self-validation and dependability. The architecture builds on the traditional autonomic architecture (denoted as the *AutonomicController* (AC) component). Other components include *ValidationCheck* (VC –which is integrated with the decision-making object of the controller to validate all *AutonomicController* decisions) and *DependabilityCheck* (DC) component, which guarantees stability and reliability after validation. The DC component works at a different time scale, thus overseas the finer-grained sequence of decisions made by the AC and VC.

The AC component (based on, e.g., MAPE logic, IMD framework, etc.) monitors the managed sub-system for context information and takes decision for action based on this information. The decided action is validated against the system's goal (described as policies) by the VC component before execution. If validation fails, (e.g., policy violation) it reports back to the AC otherwise the DC is called to ensure that outcome does not lead to, for example, instability in the system.

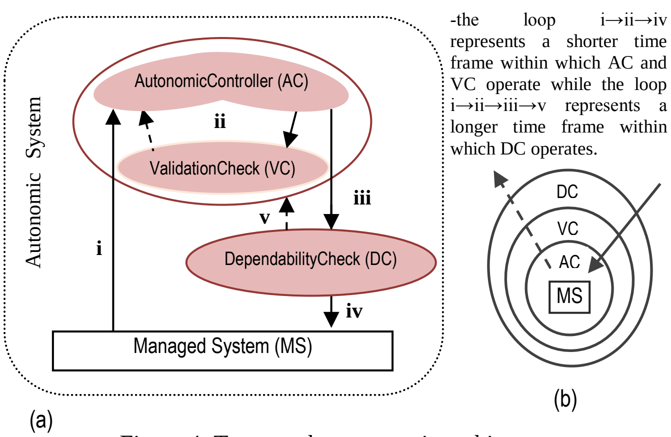


Figure 4: Trustworthy autonomic architecture

The *DependabilityCheck* component comprises of other sub-components, which makes it possible to be adapted to address different challenges. This feature makes the architecture generic and suitable to address even evolving autonomic capability requirements. For instance, in [22], the architecture is adapted to address interoperability challenges in complex interactions between AMs in multi-manager scenarios. *Predictive* component is one example of the *DependabilityCheck* sub-components that allows it to predict the outcome of the system based on the validated decision. The *DependabilityCheck* either prevents execution and sends feedback in form of some calibration parameters to the *AutonomicController* or calls the actuator to execute the validated decision.

A. *Overview of the TAArch architecture components*

This section presents the TAArch architecture in a number of progressive stages addressing it in an increasing level of detail. First, the self-management process is defined as a *Sense–Manage–Actuate* loop where *Sense* and *Actuate* define *Touchpoints* (the autonomic manager's interface with a managed system) and *Manage* is the embodiment of the actual autonomic self-management. Figure 5 is a detailed representation of the architectural framework.

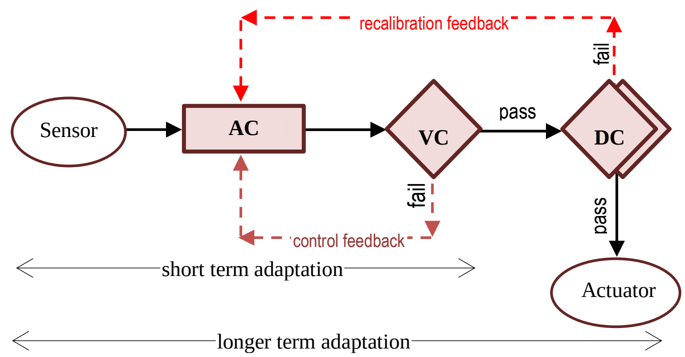


Figure 5: Detailed structure of the TAArch framework.

Traditionally, the *AutonomicController* (AC) senses context information, decides (following some rules) on what action to take and then executes the action. This is the basic routine of any AM and is at the core of most of the autonomic architectures in use today (Figure 3). At this level the autonomic unit matters but the content of the unit does not matter much, i.e., it does not matter what autonomic control logic (e.g., MAPE, IMD, etc.) that is employed so long as it provides the desired autonomic functionalities. This means that the AC component can be configured according to any autonomic control logic of choice making the framework generic as it is not tied to any one control logic. Basically, the AC component introduces some smartness into the system by intelligently controlling the decision-making of the system. Once an action is decided, following detailed analysis of context information, the decision is passed on for execution. This is at the level of sophistication defined by the autonomic architecture life-cycle level 1 (Figure 3 (i) and (ii)). So, the AC component of the TAArch framework provides designers the platform to express rules that govern target goal and policies that drive decisions on context information for system adaptation to achieve the target goal.

But, the nature of ASs raises one significant concern; input variables (context info) are dynamic and (most times) not predictable. Although rules and policies are carefully and robustly constructed, sensors (data sources) sometimes do inject rogue variables that are capable of thwarting process and policy deliberations. In addition, the operating environment itself can have varying volatility –causing a controller to become unstable in some circumstances. Thus, a mechanism is needed to mitigate behavioural (e.g., contradiction between two policies, goal distortion, etc.) and structural (e.g., illegal structure not conforming to requirement, division by zero, etc.) anomalies. This is where the *ValidationCheck* (VC) component comes in. It should be noted that AC will always decide on action(s) no matter what the input variable is. Once the AC reaches a decision, it passes control to the VC, which then validates the decision and passes it on for execution. If the check fails, VC sends *control feedback* (CF) to AC while retaining previous passed decision. A control feedback is more of an inhibition command that controls what actions are and are not allowed by the manager. This can be configured according to deployment requirements. In a nutshell, the VC, while focusing on the goal of the system, deploys self-validation mechanisms to continuously perform self-validation of the manager's behaviour and configuration against its behavioural goals and also reflects on the quality of the manager's adaptation behaviour. Again, the nature and level of test is entirely user-defined. So, the VC is a higher level mechanism that oversees the AM to keep the system's goal on track. The ultimate concern here is to maintain system goal adhering to defined rules, i.e., adding a level of trust by ensuring that target goal is reached only within the boundaries of specified rules. It is then left for designers to define what constitute validation '*pass*' and validation '*fail*'. Actual component logic are application specific but some examples in literature include fuzzy logic [24], reinforcement learning [23], etc. This is at the level of sophistication defined by the autonomic architecture life-cycle level 2 (Figure 3 (iii) and (iv)).

But in real life we understand that despite the AM taking legitimate decisions within the boundaries of specified rules, it is still possible to have overall system behavioural inconsistencies. That is, a situation where each individual decision could be correct (by logic) and yet the overall behaviour is wrong. This kind of situation where the manager erratically (though legally) changes its mind, thereby injecting oscillation into the system, could be a major concern especially in large scale and sensitive systems. This is beyond the level of current consideration in the state of practice (Figure 3). Therefore, it is necessary to find a way of enabling the AM to avoid unnecessary and inefficient change of decisions that could lead to oscillation. This task is handled by the DC component. It allows the manager change its decision (i.e., adapt) only when it is necessary and safe to do so. Consider a simple example of a room temperature controller in which, it is necessary to track a dynamic goal –a target room temperature. The AM is configured to maintain the target temperature by automatically switching heating ON or OFF according to the logic in (1). A VC would allow any decision or action that complies with this basic logic.

$$\begin{aligned} &IF\ RoomTemp < TargetTemp\ THEN\ ON\_Heating \\ &IF\ RoomTemp > TargetTemp\ THEN\ OFF\_Heating \end{aligned} \tag{1}$$

With the lag in adjusting the temperature the system may decide to switch ON or OFF heating at every slight tick of the gauge below or above target (when room temperature is sufficiently close to the target temperature). This may in turn cause oscillation, which can lead to undesirable effects. The effects are more pronounced in more sensitive and critical systems where such changes come at some cost. For example, a datacentre management system that erratically switches servers between pools at every slight fluctuation in demand load is cost ineffective. Actual component and sub-component logic are user-defined. One powerful logic example, as explained in Section II, for implementing the DC component is the dead-zone (DZ) logic [16]. DZ logic has been shown to offer a reliable means of achieving self-stabilisation, dependable systems and TAC.

The DC component may also implement other sub-components like Prediction, Learning, etc. This enables it to predict the outcome of the system and to decide whether it is safe to allow a particular decision or not. An example sub-component logic is Trend Analysis (TA) logic. TA logic identifies patterns within streams of information supplied directly from different sources (e.g., sensors). By identifying trends and patterns within a particular information, (e.g., spikes in signal strength, fluctuation in stock price, rising/falling trends etc.) the logic enables the AM to make more-informed control decisions and this has the potential of reducing the number of control adjustments and can improve overall efficiency and stability. Also, the analysis of recent trends enables a more accurate prediction of the future. With TA, managers can base decisions on a more-complete view of system behaviour. The usage and importance of TA are discussed in more detail in [16].

So after validation phase, the DC is called to check (based on specified rules) for dependability. DC avoids unnecessary and inefficient control inputs to maintain stability. If the check passes, control is passed to the Actuator otherwise a *recalibration feedback* (RF) is sent to AC. An example of RF is dynamically adjusting (or retuning) the DZ boundary width (explained later) as appropriate. The RF enables the manager to adjust its behaviour to maintain the level of required trust. So, while VC looks at the immediate actions, DC takes a longer term view of the manager's behaviour over a certain defined time interval. A particular aspect of concern, though, is that for dynamic systems the boundary definition of DZ may itself be context dependent (e.g., in some circumstances it may be appropriate to allow some level of changes, which under different circumstances may be considered destabilising). This concern is taken into consideration when defining such boundaries.

So the current state-of-the-art of autonomic architecture suffices for short term adaptation. To handle longer term frame adaptation, e.g., cases where continuous validation fails to guarantee stability and reliability, requires a robust autonomic approach. This robust autonomic approach is what the proposed TAArch offers. Consider the whole architecture as a nested control loop (Figure 4 (b)) with AC the core control loop while VC and DC are intermediate and outer control loops, respectively. In summary, a system, no matter the context of deployment, is truly trustworthy when its actions are continuously validated (i.e., at run time) to satisfy set requirements (system goal) and results produced are dependable and not misleading.

IV. IMPLEMENTATION AND EMPIRICAL ANALYSIS

To demonstrate the feasibility and practicability of the new architecture, this section presents an implementation and simulation analysis of the TAArch architecture using a datacentre case example scenario. This analysis is a complex and robust implementation of TAArch demonstrated in a resource allocation scenario, which models basic datacentre resource allocation management. Although the demonstration uses a datacentre scenario, which though offers a way of efficiently managing complex datacentres, the application of TAArch can be widespread. In other words, although a datacentre is used to demonstrate the functionalities of the proposed architecture, it is not limited to this scenario. The datacentre model represents a very simple datacentre scenario where the simulation focuses on the efficiency and dependability of resource request and allocation management rather than other vast areas of datacentre, e.g., security, power, and cooling etc. So the purpose of the experiments is to demonstrate the applicability and performance of the proposed architecture and not to investigate datacentres themselves. However, the datacentre is chosen as implementation scenario because its many dimensions of complexity and large number of tuning parameters offer a rich domain in which to evaluate a wide range of techniques, tools and frameworks.

In this example, detailed experiments are designed to analyse three different systems based on three different autonomic architectures. The first system, comprising of only AC component, is based on the traditional architecture represented by level 1 (Figure 3 (i) and (ii)) of the autonomic architecture life-cycle. This system will be referred to as sysAC. The second system, comprising of both the AC and VC components, is based on the current level of practice represented by Figure 3 (iii) and (iv). This system will be referred to as sysVC. The third and TAArch-based system, referred to as sysDC, comprises of all three (AC, VC, and DC) components. This system falls within the representation of level (v) of Figure 3. The purpose of this implementation is to illustrate how powerful and robust the TAArch framework is when compared to existing frameworks.

A. *Scheduling and Resource Allocation*

Several research, e.g., [25][26][27], have proposed scheduling algorithms that optimise the performance of datacentres. In a utility function based approach, Das *et al.* [25] are able to quantify and manage trade-offs between competing goals such as performance and energy consumption. Their approach reduced datacentre power consumption by up to 14%. Other works that have resulted in improved performance and resource utilisation by proposing new scheduling algorithms include [26], which focuses on the allocation of virtual machines in datacentre nodes and [27], which uses a '*greedy resource allocation algorithm*' that allows distributing a web workload among different servers assigned to each service. Our work, on the other hand, does not propose any new scheduling algorithm for efficient utilisation of datacentre resources; however, it uses basic resource allocation technique to model the performance of datacentre autonomic managers in terms of the effectiveness of resource request and allocation management.

Let us consider the model of the datacenter used in this experimentation in detail, (in terms of scheduling and request services). The datacentre model comprises a pool of resources S_i (live servers), a pool of shutdown servers $\check{S}_i$ (ready to be powered and restored to S_i as need be), a list of applications A_j, a pool of services $\underset{..}{U}$ (a combination of applications and their provisioning servers), and an autonomic manager (performance manager *PeM*) that optimises the entire system. A_j and S_i are, respectively, a collection of applications supported (as services) by the datacentre and a collection of servers available to the manager (*PeM*) for provisioning (or scheduling) available services according to request. As service requests arrive, *PeM* dynamically populates $\underset{..}{U}$ to service the requests. $\underset{..}{U}$ is defined by equation (2):

$$\underset{..}{U} = \begin{cases} A_1: (S_{11}, S_{12}, S_{13}, \ldots, S_{1i}) \\ A_2: (S_{21}, S_{22}, S_{23}, \ldots, S_{2i}) \\ \ldots \;\ldots \;\ldots \;\ldots \;\ldots \;\ldots \\ A_n: (S_{n1}, S_{n2}, S_{n3}, \ldots, S_{ni}) \end{cases} \tag{2}$$

Where A_i: ($S_i \ldots S_n$) means that ($S_i \ldots S_n$) servers are currently allocated to Application A_i and n is the number of application entries into $\underset{..}{U}$. (2) indicates that a server can be (re)deployed for different applications. All the servers i in S_i are up and running (constantly available –or so desired by *PeM*) waiting for (re)deployment. The primary performance goal of *PeM* is to minimise oscillation and maximise stability (including just-in-time service delivery to meet service level achievement target) while the secondary performance goal is to maximise throughput.

Service (application) requests arrive and are queued. If there are enough resources to service a particular request then it is serviced otherwise it remains in the queue (or may eventually be dropped). The manager checks for resource availability and deploys server(s) according to the size of the request. The size of application requests and the capacity of servers are defined in million instructions per second (MIPS). In this report '*size*' and '*capacity*' are used interchangeably and mostly would refer to MIPS i.e., the extent of its processing requirement. When a server is deployed it is placed in a queue for a time defined by the variable *ProvisioningTime*. This queue simulates the time (delay) it takes to load or configure a server with necessary application. Recall from Equation (2) that any server can be (re)configured for different applications and so servers are not pre-configured. Servers are then '*Provisioned*' after spending *ProvisioningTime* in the queue. The provisioning pool is constantly populated as requests arrive. Now as a result of the lag between provisioning time and the rate of request arrival or as a result of some unforeseen process disruptions, some servers do overshoot their provisioning time and thereby left redundant in the queue. This can be addressed by the manager, depending on configuration, to reduce the impact on the whole system. As requests are fully serviced (completed) servers are released into the server pool and redeployed. Note that service level achievement (SLA) is calculated based on accepted requests. Rejected or dropped requests are not considered in calculating SLA. The essence of the request queue is to allow the manager to accept requests only when it has enough resources to service them. Service contract is entered only when requests are accepted. So the manager could look at its capacity (in terms of available resources), compare that with the capacity requested and say '*sorry I haven't got enough resources*' and reject or drop the request. This whole process goes on and the manager manages the system to the level of its sophistication. This process is explained in Appendix A.

A basic system without any form of smartness can barely go far before the whole system is clogged due to inefficient and unstructured resource management. The level to which any autonomic manager can successfully and efficiently manage the process defined above depends on its level of sophistication. For us this largely depends on how each manager is wired (in terms of architecture) and not necessarily the scheduling algorithm or actual component logic used. For example, two managers, differently wired, may employ the same scheduling algorithm but achieve different results. Results here may be looked at in terms of, say, '*with such level of available resources how many requests were successfully serviced*'. These are the kind of considerations in the following experiments where three differently wired autonomic managers are analysed.

B. Experimental Design, Workload and Parameters

The experiments are designed and implemented using the TAArch application (Appendix A). This application is developed using the C# programming language. The scope of the experiments focuses on the performance of datacentre autonomic managers in resource request and allocation management activities under varying workloads. Although some workload parameters are sourced from experimental results of other research, e.g., [28][29][30], the designed experiments allow for the tailoring of all parameters according to user preferences. Simulations are designed to model several options of real datacentre scenarios. So, depending on what is being investigated the user can design individual scenarios and set workloads according to specific requirements.

The result of every simulation analysis is relative to the set of workload or parameter set used, which configure the specific application instance. The parameter set used for the datacentre model analysis here are classified into *internal* and *external* variables. Internal variables are those variables that do not change during run-time, e.g., the capacity of a server. External variables, on the other hand, are those that can change in the cause of the simulation, e.g., the rate at which requests arrive. External variables are usually system generated and are always unpredictable. The experimental design has the capacity for heterogeneous workload representation. That means that even the internal variables can be reset before simulation begins thereby offering the possibility of scaling to high/low load to suit user preferences (see Appendix A). The range of value options for most of the variables reflects the experimental results of other research especially [28][29][30]. Note that the following variables are used with the C# application that has been designed to simulate the datacentre model and run the stated experiments.

- **Internal Variables**

Below is the list of internal variables used in this experiment. Some of the variables used are specific to this experiment while some are general datacentre variables.

- ***server.sCapacity***:

This represents the service capacity of each server and for the purposes of the experiments here all servers are assumed to be of equal capacity. Server capacity (size) is measured in MIPS.

- ***RetrieveRequestParam***:

Tuning parameter indicating when to start shutting services (this simulates service request completion) –at which point some running requests are closed as completed. This value is measured as percentage of number of servers in use and has been restricted to value between 0.1 and 0.3. The margin 0.1 – 0.3 (representing 10 to 30%) is used because experiments show that it is the safest margin within which accurate results can be guaranteed. The datacentre is not completely settled below 10% and beyond 30% scenarios with low number of servers will yield inaccurate results. The higher the value of *RetrieveRequestParam* the earlier the start of request completion.

- ***RetrieveRate***:

Indicates rate at which requests are completed once simulation for service request completion is initiated. Value is relative to rate of request arrival – e.g., if value is 5, then it means service request completion is five times slower than rate of service request.

- ***BurstSize***:

Indicates how long the user wants the burst (injected disturbance) to last. This value is measured in milliseconds. Burst is a disturbance introduced by the user to cause disruption in the system. This alters the smooth running of the system and managers react to it differently. Often times injecting a burst disorientates the system. The nature of this disruption is usually in the form of sudden burst or significant shift in the rate of service request.

- ***ServerProvisioningTime***:

Indicates how long it takes to load or configure a server with an application. This is relative to the rate of request arrival -it is measured as half the rate of request arrival, e.g., the value of 3 will translate to 1.5 of rate of request arrival.

- ***ServerOnTime***:

Indicates how long it takes a server to power on. This is relative to the rate of request arrival -it is *ServerProvisioningTime* + 1.

- ***RequestRateParam***:

A constant used to adjust the possible range of request rate. The user of the TAArch Application (Appendix A) can set request rate according to preference but this preference may not be accommodated within the available rate range. For example, if the least available rate is 1 request/second and the user wishes to use 2 requests/second, the *RequestRateParam* parameter can be used to extend the available range. A higher value increases the range for a lower rate of request arrival.

- **External Variables**

Below is the list of external variables used in this experiment. Recall that external variables, also known as dynamic variables, are those variables that are fed into the system during run-time either as system generated (dynamic sensitivity to contextual changes) or human input (through external touch-points). Some of the variables used are specific to this experiment while some are general datacentre variables.

- ***DZConst***:

DZConst is the tuning parameter the manager uses to dynamically adjust dead-zone boundaries. The dead-zone boundary is also known as DZWidth. Because this variable has significant effect on the system, it is suggested that the initial value be set at 1.5. The manager usually adjusts this value dynamically and there is also a provision to manually adjust the value during run time.

- ***AppSize***:

The application size variable represents the size or capacity of a service request (request for an application). In the experiments that follow, except otherwise changed, all applications are initially assumed to be of the same size. There are touch-points to dynamically change this value. The application size variable is measured in MIPS.

- ***RequestRate***:

This variable also referred to as rate of service request or rate of request arrival is the measure of the frequency of service request. This is in terms of the number of requests recorded per unit of time. In real systems, this can be calculated as an average for all services (applications) or for individual services. In [28], for example, *RequestRate* values are calculated for each service and are presented in *requests/day*. The experiments of this work take an average of *RequestRate* for all services and represent values as *requests/second*.

- ***BurstInterval***:

The burst interval variable defines the interval at which bursts are injected into the system during the simulation. This is specific to the experimental application and is dependent on what the user wants to investigate. Usually bursts are introduced once at a specific time or several at random times.

The experimental workload is flexible in that all variables can be scaled to suit user's workload (high or low) requirements. Every experiment has a detailed workload outline used as shown in the following experiments.

C. Manager Logic

Manager logic details the individual control logic employed by each of the managers in order to achieve the performance goal. This explains the logical composition of each manager. The three autonomic managers track the life-cycle of autonomic architecture as presented in Figure 3. *sysAC* represents the *AutonomicControler* level based manager while *sysVC* represents the *ValidationCheck* level based manager. *sysDC* represents the *DependabilityCheck* level based manager and this conforms to TAArch architecture.

The primary goal of the AM (also referred to as the performance manager –*PeM*), represented by each of *sysAC*, *sysVC*, and *sysDC*, is to ensure that the system remains stable under almost all perceivable operating and contextual circumstances and is capable of achieving desired and dependable results within such circumstances (i.e., over the expected range of contexts and environmental conditions and beyond). The secondary goal is to maximise throughput.

- ***sysAC***

This manager implements the basic autonomic control logic. Structurally based on Figure 3 (ii), the manager receives requests and allocates resources accordingly. The basic allocation logic here is to deploy a server whenever capacity offset (i.e., excess capacity of running servers –these are used to service new requests) is less than the current capacity of a single request. This is known as the *DecisionBoundary*. This is depicted, for example, as:

```
if (app1ACOffset < app1.appCapacity)
 {        <...deploy server...>   }
```

Where

```
app1ACOffset = app1ACAvailableCapacity -
          app1ACRunningCapacity;
```

sysAC has no additional intelligence. For example, decisions are not validated and the manager does not consider the rate at which system behaviour crosses the *DecisionBoundary*. As long as boundary conditions are met, the manager executes appropriate decisions.

- ***sysVC***

This manager shows a higher level of intelligence than *sysAC*. One aspect of validation here is to check the performance of the manager in terms of correctness. The manager does not start a job that cannot be completed –i.e., at every *DecisionBoundary* the manager checks to make sure that it has enough resources to service a request. Where this is not the case (meaning the check has *failed*), the manager rejects the request and updates itself. The manager has a limit to which it can allow capacity deficit expressed as:

```
else if (app1VCOffset <= (0 - app1.appCapacity))
     {
         DroppedRequestCountVC += 1;
     }
```

So, in addition to the basic control and resource allocation logic of sysAC, sysVC carries out a validation of every allocation decision. Validation here is in terms of behavioural (e.g., starting a job only when there are enough capacity to complete it) and structural (e.g., avoiding initiating provisioning when server pool is empty i.e., `listViewServer.Items.Count = 0`) correctness.

sysVC is within the representation of current stages of autonomic architecture life-cycle presented in Section II as Figure 3 (iii) and (iv). Beyond the level of validation, *sysVC* exhibits no further intelligence.

- ***sysDC***

sysDC performs all the activities of the *sysAC* and *sysVC* managers with additional intelligence. The manager looks at the balance of cost over longer term and retunes its configuration to ensure a balanced performance. For example, the manager implements dead-zone (DZ) logic on decision boundaries. Firstly, the dead-zone boundaries (upper and lower bounds), for example, are calculated as:

```
App1.DZUpperBound = (app1.appCapacity + (app1.appCapacity * DZ.DZConst));      (3)
App1.DZLowerBound = (app1.appCapacity - (app1.appCapacity * DZ.DZConst));
```

Note: the size of DZ boundary depends on the nature of the system and data being processed. For example, in fine-grained data instance, where small shifts from the target can easily tip decisions –sometimes leading to erratic behaviour, the DZ boundary is expected to be small and closely tracked to the target value. However, in other cases as in this experiment, the DZ boundary cannot be as closely tracked to the target value. Here the target value (*DecisionBoundary*) is defined by capacity Offset (see (7) later) and this is used by the AM to decide whether or not to deploy a server. And because Offset is populated in `serverCapacity` and depleted in `appCapacity` (i.e., the difference between available and requested capacity) any behaviour shift across the decision boundary (on either side of the boundary) is in excess of `appCapacity`. This means that fluctuations around the decision boundary are usually in multiples of `appCapacity` and to handle erratic behaviour around *DecisionBoundary* the AM will need to take `appCapacity` into consideration when calculating DZ boundaries. This explains the boundary size calculation of (3). Offset is positive when there is excess capacity than required and negative when there is a shortfall. Also, sample simulation results show that smaller sizes of dead-zone boundary have no effect on the system behaviour.

Secondly, the zone areas are defined as follows (two zones are defined with one on either side of the *DecisionBoundary* –see Figures 8 and 9):

```
if (app1DCOffset < app1.appCapacity)
{
        App1.SystemBehaviour = "IsInDeployZone";
}
else
{
        App1.SystemBehaviour = "IsNotInDeployZone";
}
```

Then stability is maintained by persisting the behaviour (*DecisionBoundary* outcome) of the system across the zones as follows:

```
if (app1DCOffset >= app1.appCapacity)
{ App1.SystemBehaviour = "IsNotInDeployZone"; }

if   ((App1.SystemBehaviour  ==  "IsInDeployZone")  &&
      (app1DCOffset < App1.DZUpperBound))
{ App1.SystemBehaviour = "IsInDeployZone"; }
else
{ App1.SystemBehaviour = "IsNotInDeployZone"; }

if  ((App1.SystemBehaviour  ==  "IsNotInDeployZone")  &&
      (app1DCOffset > App1.DZLowerBound))
{ App1.SystemBehaviour = "IsNotInDeployZone"; }
else
{ App1.SystemBehaviour = "IsInDeployZone"; }
```

Thus, the *DecisionBoundary* in *sysAC* and *sysVC*, which is `(app1DCOffset < app1.appCapacity)` now becomes `(App1.SystemBehaviour == "IsInDeployZone")` in *sysDC*. The AM dynamically changes the `DZ.DZConst` value between three values of 1, 1.5 and 2. By doing this the manager is sensitive to its own behaviour and proactively regulates (retunes) its decision pattern to maintain stability and reliability.

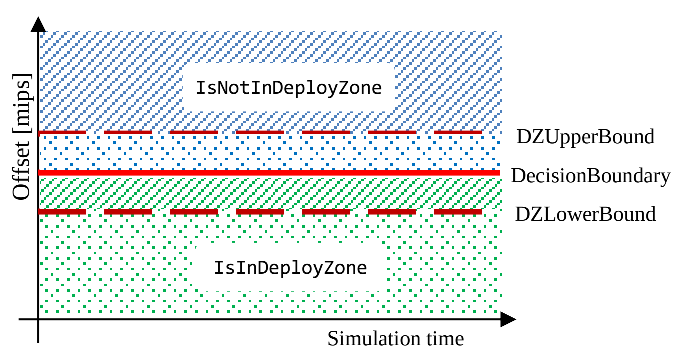


Figure 6: Dead-zone logic implemented by SysDC.

In Figure 6, the area shaded in green represents the '*IsInDeployZone*', which means the manager should deploy a server while the area shaded in blue represents the '*IsNotInDeployZone*', which means the manager should not deploy a server. Likewise, the dotted shade pattern represents the '*IsInDeployZone*' while the diagonal shade pattern represents the '*IsNotInDeployZone*'. As shown, if, for example, the system behaviour falls within the '*IsNotInDeployZone*' area, the manager persists the action associated to this zone until system behaviour falls below the '*DZLowerBound*' boundary at which point the action associated to the '*IsInDeployZone*' area is activated. This way the AM is able to maintain reliability and efficiency. The AM also retunes its behaviour (as explained earlier) by adjusting *DZWidth* if fluctuation is not reduced to an acceptable level. Thus, three behaviour regions (in which different actions are activated) are defined; 'upper region' (*IsNotInDeployZone* with 'DO NOT DEPLOY SERVER' action), 'lower region' (*IsInDeployZone* with 'DEPLOY SERVER' action), and 'in DZ' (within the DZWidth with either of the two actions). It is important to note, as shown in Figure 6, that within the DZ boundary (i.e., the 'in DZ' region), either of the actions associated to '*IsInDeployZone*' and '*IsNotInDeployZone*' areas could be maintained depending on the 'current action' prior to deviation into the 'in DZ' region. So actions activated in the 'upper region' and 'lower region' are respectively persisted in the 'in DZ' region. This is further explained in Figure 7, which shows the resultant effect of the DZ logic in terms of what zone action is activated per time.

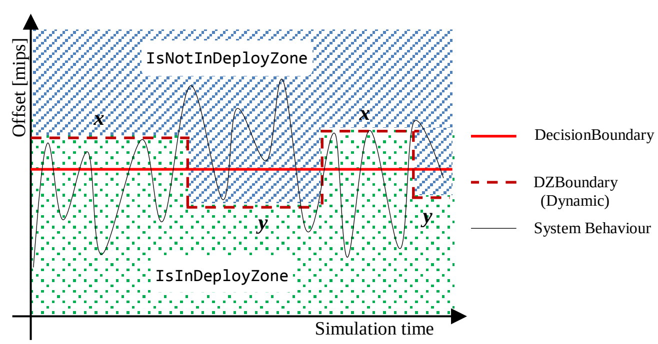


Figure 7: Illustration explaining actual performance effect of DZ logic.

Figure 7 explains what happens in Figure 6. As system behaviour fluctuates around decision boundary, the manager dynamically adjusts the DZ boundary to mitigate erratic adaptation. As shown, minor deviations across the DecisionBoundary do not result in decision (or action) change. In this case (Figure 7) actions for `IsInDeployZone` and `IsNotInDeployZone` are persisted at states *x* and *y* respectively despite system behaviour crossing the decision boundary at those state points.

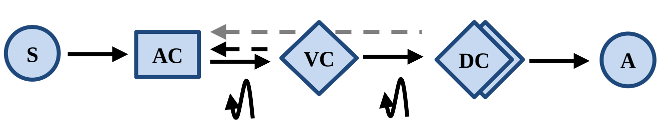


Figure 8: Structural representation of sysDC.

Figure 8 is a representation of the next level of sophistication in autonomic architecture life-cycle required to ensure dependability. This is presented in Section II as Figure 3 (v).

To illustrate the overall operation of the DZ logic, a simple numeric example is given: Let us consider a simple use-case example in which a room temperature controller is set to maintain temperature at 20^0C: The AM is configured to turn ON heating when room temperature falls below the target temperature (20^0C) and to turn OFF heating otherwise. If, for example, the room temperature keeps fluctuating between 19^0C and 21^0C the manager will as well fluctuate with its decisions (i.e., erratic behaviour of frequently turning heating ON and OFF). This situation is undesirable and can be enormously costly in crucial systems. To mitigate this situation, the manager can implement DZ logic with a DZLowerBound of 19^0C and DZUpperBound of 21^0C. This will allow the manager to turn off heating only when room the temperature rises above 21^0C and to turn on heating only when it falls below 19^0C. Putting this in the context of (20) means that, e.g.:

```
DZUpperBound = (20 + (20 * 0.05))
DZLowerBound = (20 – (20 * 0.05))
```

This will calm the erratic behaviour of the AM. However, if the erratic behaviour does not drop to an acceptable level the manager can further retune itself by increasing *DZConst* by multiples of 0.05 (e.g., *DZConst* += 0.05). If on the other hand the AM discovers that it is not making decisions frequently enough, (i.e., the room is getting too cold or too hot) it can retune its behaviour to increase its rate of decision-making by reducing the DZ boundaries (e.g., *DZConst* –= 0.05). So the AM retunes itself by dynamically adjusting the DZ boundaries using (*DZConst* $\pm$= 0.05) as appropriate. It is important to note that the average of the DZ boundaries is equal to the target goal – e.g., the average of 19^0C and 21^0C is 20^0C, which is the target temperature.

D. Simulation Scenarios and Metrics

In the following simulations to analyse the performances of the three systems (*sysAC*, *sysVC* and *sysDC*), four simulation scenarios are used. The scenarios are presented in Table I. The user of the TAArch application can define further scenarios as required.

Table I: Resource allocation simulation scenarios

Scenario	Description	Metrics
Scenario 1	Basic simulation with uniform request rate and application size	SLA Delay cost Server deployment rate Optimum provisioning (Offset analysis)
Scenario 2	Basic simulation with uniform request rate and varying application sizes	
Scenario 3	Uniform application size with burst injected at a particular time in the simulation	
Scenario 4	Varying application sizes with inconsistent request rate	

Scenario 1: In scenario 1, all parameters are kept constant except those (e.g., *DZConst*) that may need dynamic tuning by the manager as need arises. This scenario gives a default view of the behaviour of the managers under normal condition. Under this scenario of normal condition, it is expected that all managers will behave significantly closely.

Scenario 2: This scenario creates a condition where the managers will have to deal with irregular sizes of service request. This leads to contention between applications –huge applications will demand huge resources thereby starving smaller applications. Performance analysis here will include individual application analysis. Request rate is kept constant so that the effect of varying application sizes could be better analysed.

Scenario 3: In this scenario, request rate and application size are kept constant while burst is injected at a chosen time (*SimulationTime*) in the simulation. This is similar to Scenario 1 just that a sudden and unexpected disruption (burst) is injected into the system. This will measure the robustness of the AMs in adhering to the goal of the system. The impact of the burst is relative to the size of the burst (*BurstSize*).

Scenario 4: This is the most complex scenario with resource contention and two instances of burst injection. This scenario creates the combined effect of Scenarios 2 and 3 put together. Request sizes vary leading to resource contention and request rate is highly erratic. Inconsistent request rate can also lead to '*flooding*', which also is a kind of burst. *Flooding* is a situation where the system is inundated with requests at disproportionate rate.

All metrics are mathematically defined giving the reader a clear picture of the definition criteria should they wish to replicate this experiment.

SLA: Service level achievement is the ratio of provided service to requested service. It measures the system's level of success in meeting request needs. Note that requests and services are not time bound so the time it takes to complete a request does not count in this regard. The metric is defined as:

$$SLA = \begin{cases} \frac{ProvisionedCapacity}{RequestedCapacity} & \text{(i)} \\ \frac{AvailableCapacity}{RunningCapacity} & \text{(ii)} \end{cases} \tag{4}$$

Where *ProvisionedCapacity* is the total deployed server capacity (excluding those in queue and including those already reclaimed back to the pool) and Requeste*dCapacity* is the total size of request (including completed requests). *AvailableCapacity* is *ProvisionedCapacity* minus capacity of reclaimed servers (*ReclaimedCapacity*) while *RunningCapacity* is the total size of request (excluding completed requests). In (4), (i) is more of a whole picture consideration –considering the entire capacity activities of the system while (ii) takes a real time view of the system – tracking to the minute details of the system with delay, completed requests and reclaimed server effects all considered. The reference value for SLA is 1 indicating 100%. Values above 1 indicate over-provisioning while values under 1 indicate shortfall. Optimum provisioning is achieved at close proximity to 1.

Delay cost: Delay cost can be calculated in many different ways as the cost can be influenced by many delay contributors. In this instance, delay cost is defined as the cost (in capacity) as a result of the delay experienced by the servers. This delay affects the completion time of service requests. This is mathematically represented as:

$$\begin{aligned} DelayCost &= \frac{(DeployedCapacity - ProvisionedCapacity)}{DeployedCapacity} \\ &= \frac{ProvisioningCapacity}{DeployedCapacity} \end{aligned} \tag{5}$$

ProvisioningCapacity is the capacity of servers in queue while *DeployedCapacity* is the total capacity of all deployed servers. The lower value of delay cost means the better performance of the system.

Deployment Rate: Server (re)deployment rate is the ratio of server deployment to service request. It measures the frequency at which managers deploy servers with regards to the nature of requests. This is mathematically represented as:

$$DeploymentRate = \frac{DeployedCapacity}{(RequestedCapacity - CompletedCapacity)} \tag{6}$$

The lower value of deployment rate means the better performance of the system translating to better maximisation of throughput.

Optimum provisioning: This metric is also an offset analysis. It indicates whether and when the manager is over or under provisioning. This is also known as efficiency calculation. Offset is calculated as:

$$Offset = AvailableCapacity - RunningCapacity \tag{7}$$

Under normal circumstances, average offset is not expected to fall below zero. The system is optimally provisioning when offset falls between zero and the average capacity of all applications. The closer to zero the offset value is, the better the performance of the system.

Note that, for all metrics, low or high values do not always necessarily translate to better performance. It is not usually realistic for the supposed better manager to always outperform the other managers. There are times when the manager underperforms and usually there may be a tradeoff of some kind that explains the situation.

E. Experimental Results

Results are presented and analysed according to simulation scenarios. For precise results, ten different simulations of each Scenario are performed and results presented are based on average of these ten simulations. For each of the ten simulations, the parameters used are presented. It is important to note the workload and parameters used for individual simulations as results will largely depend on those.

Scenario 1: Basic simulation with uniform request rate and application size

Table II is a collection of major parameters used in this scenario. The number of requests and the distribution of those requests amongst applications differ with each AM as they are dynamically generated and unpredictable. This does not distort the results as analysis is based on system-wide performance and not on individual application performance.

Table II: Scenario 1 simulation parameters

Parameter		Value
# of servers		300
# of applications		4
Request rate		1 req/sec
Application capacity (MIPS)		20000
Server capacity (MIPS)		40000
Internal variables	RetrieveRate	5x
	RequestRateParam	10
	RetrieveRequestParam	0.2
	ServerProvisioningTime	3 (1.5 sec)
Managers (*sysAC*, *sysVC* & *sysDC*)		PeM
DZConst		1.5

In every simulation, there are 300 servers of 40000 MIPS capacity each. This means there is a total of initial 12000000 MIPS to share between requests for four applications (App1, App2, App3, and App4). Reclaimed servers are later added to this available capacity. If the total requested capacity is higher than the total provisioned capacity, the unused server list will be empty (leaving the manager with a deficit of outstanding requests without resources to service them) and the datacentre is overloaded. So the simulation stops whenever any manager runs out of resources (i.e., when the unused server list of any manager becomes empty). It is necessary to stop the simulation at this point because as soon as the unused server list of a particular manager becomes empty, the *RequestedCapacity* for that manager starts piling up while *AvailableCapacity* remains at zero, which leads to continuously increasing negative *Offset*. This will lead to inaccurate assessment of the three managers (recall that all three managers are compared concurrently and it is safer to do this while all three managers are active). Also, at this point, usually, other managers may have outstanding resources and this will mean better efficiency. Table III is a number distribution of requests and services for ten simulation runs of Scenario 1. The values shown are collected at the end of each simulation, for example, it can be seen that the manager of *sysAC* has no servers left in each of the simulations while *sysVC* has a couple and *sysDC* even more. Though *sysAC* and *sysVC* are able to service almost the same number of requests, *sysVC* has outstanding server capacity and could service more requests. However, the additional smartness of *sysVC* does not always translate to better performance as highlighted in Table III (this is an example of manager interference leading to overcompensation). *sysDC* clearly outperformed the others with an average of about 36 outstanding servers out of 300 initial servers. Figures 11-14 give a breakdown of the performances.

Table III: High level performance analysis of managers over ten simulation runs of Scenario 1

Sim	unused server			serviced request			deployed server		
	AC	VC	DC	AC	VC	DC	AC	VC	DC
1	0	2	35	578	577	555	307	307	268
2	0	3	27	594	594	574	310	299	278
3	0	0	36	600	590	574	309	305	268
4	0	0	34	593	585	566	309	313	274
5	0	0	30	609	586	587	312	303	273
6	0	0	38	597	586	576	308	309	268
7	0	0	36	613	605	587	314	304	268
8	0	15	39	591	590	565	307	287	263
9	0	6	33	582	582	566	304	302	271
10	0	8	48	569	567	542	310	298	255
avg	**0**	**3.4**	**35.6**	**592.6**	**586.2**	**569.2**	**309**	**302.7**	**268.6**

The difference between requested capacity and provisioned capacity (or in real time analysis, running capacity and available capacity) is known as *Offset*. Where offset is close to zero, the difference with respect to running and available MIPS is low and the AM is therefore very efficient. When offset is much greater than or much less than zero, the AM is over-provisioning or under-provisioning respectively and is very inefficient. The AMs are designed to have a window of 'optimum provisioning' defined by the interval ($0 \leq Offset \leq AvgAppCapacity$), which means that the AM are configured to maintain *AvailableCapacity* of up to average *appCapacity* for just-in-time provisioning. However, AM efficiency is defined by its ability to maintain *Offset* as close as possible to zero. Figure 9 shows the efficiency analysis of the three managers in terms of maximising resources. This is in terms average performances of the three AMs over ten simulation runs. This means that the same scenario was run for ten times and then the average result was calculated. This gives a clearer picture and more accurate analysis of manager performance.

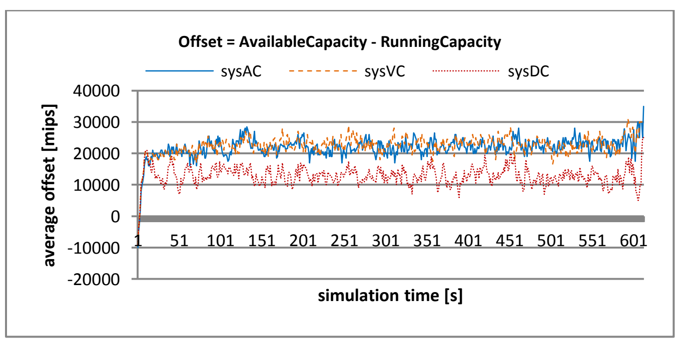


Figure 9: Manager efficiency analysis for scenario 1.

Figure 9 shows that, in terms of efficiency, *sysAC* performed significantly similar to *sysVC* with a couple of instances where *sysAC* also performed better than *sysVC*. This is as a result of over compensation introduced by the extra level of smartness in *sysVC*. The validation check of *sysVC* gives it an advantage over *sysAC* but it sometimes leads to over compensation. For example, though *sysVC* checks to ensure resource availability against resource requests, it is not adequately sensitive to erratic request fluctuation. High level of erratic request fluctuation disorientates *sysVC* (as can be seen in later scenarios where burst is injected) but this effect is naturally and dynamically handled by *sysDC*. *sysDC* takes a longer term look at the self-management effect on the datacentre and retunes its self-management behaviour. The rate at which the managers change decision, (which can indicate erratic behaviour) is indicated by the gap between the crests and troughs of the graph in Figure 9. Smaller gap indicates erratic change of decision while bigger gap indicates more persisted decision. As seen, *sysDC* has significantly more persisted decisions and this allows it to more adequately track resource availability against resource requests, which leads to more efficient performance as can be seen. Recall that optimum provisioning is defined by the ($0 \leq Offset \leq AvgAppCapacity$) interval, which in this case is between 0 and 20000 MIPS. *sysDC* clearly falls within this range, though a bit towards the 20000 border. This means that while *sysAC* and *sysVC* try to maintain *AvailableCapacity* of up to 20000 MIPS for just-in-time provisioning, *sysDC* efficiently depletes this reserve to maximise resources while at the same time maintaining the same level of performance and even better compared to the other two. This is evidently seen in the following deployment rate, SLA, and cost metrics analyses.

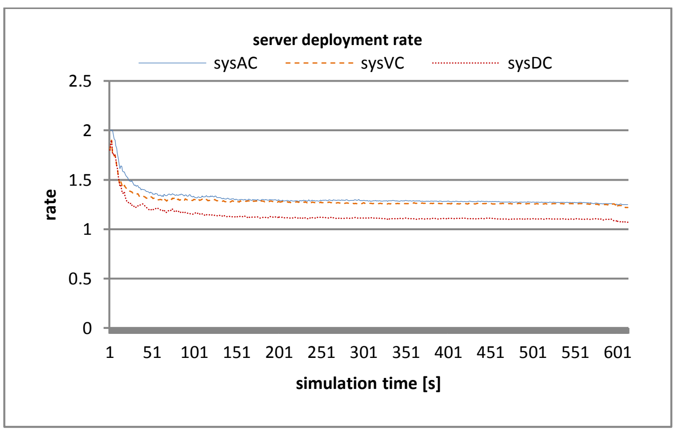


Figure 10: Server deployment rate analysis for scenario 1.

Figure 10 shows the rate at which the three AMs deploy servers as requests arrive. With the same request rate, the AMs deployed servers differently. While *sysAC* deployed the most servers, *sysDC* deployed the least servers. This explains why *sysAC* easily runs out of servers followed by *sysVC* while *sysDC* still retains a couple of unused servers (Table III). Interestingly, this does not negatively affect the performance of *sysDC* and when *sysDC* underperforms in one aspect there is usually compensation (say tradeoff) in another aspect. The lower server deployment rate of *sysDC* resulted in lower SLA value of *sysDC* (when compared to *sysAC* and *sysVC* –Figure 11) but this only keeps the value very close to the optimum value of 1, which also indicates high efficiency.

Figure 11 depicts the service levels of the three AMs with the zoomed-in inset revealing the gaps between their performances. As expected, following from the result trend above, *sysAC* and *sysVC* performed quite similarly with each outperforming the other in some places. *sysDC* on the other hand, keeps SLA as close as possible to the target goal of 1 (a perfect system would keep SLA at 1). *sysDC* has the ability to dynamically scale down unnecessary and inefficient provisioning by proactively throttling oscillation. This capability also leads to cost savings as shown in Figure 12.

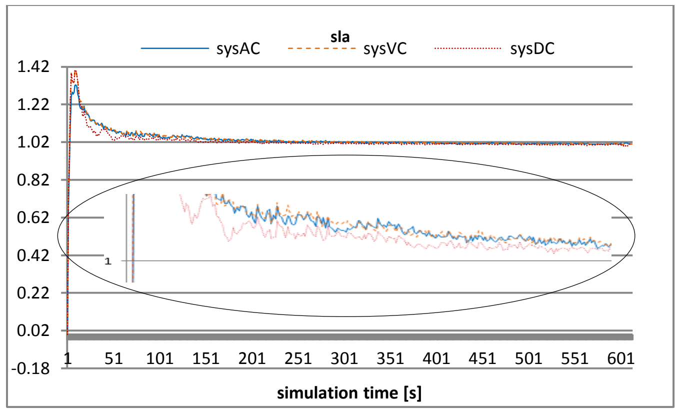


Figure 11: Service level achievement (SLA) analysis for scenario 1.

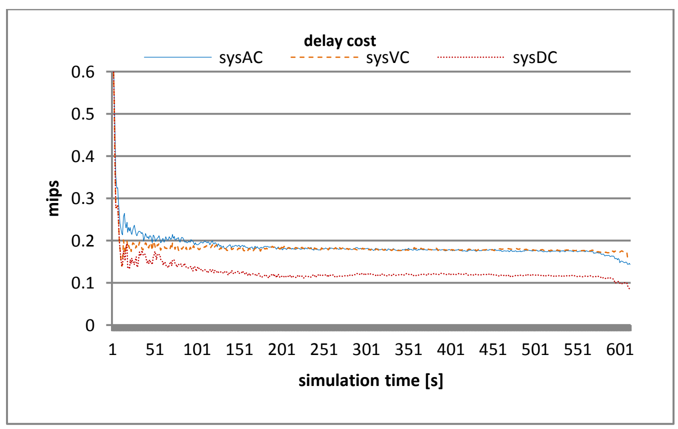


Figure 12: Delay cost analysis for Scenario 1.

The high level of deployment rate (i.e., deploying more MIPS than required) for *sysAC* and *sysVC* (Figure 10) leads to high cost (in terms of excess MIPS) of servicing individual requests. Also this means that the rate at which servers enter the provisioning queue is much higher than the rate they leave the queue. This results in an increasing number of redundant servers in the queue, which contributes to delay cost (Figure 12). Also, the number of redundant servers for *sysDC* is doubled by that of *sysAC* and *sysVC*.

The results analyses of Scenario 1 indicate that the proposed TAArch (represented by *sysDC*) has significant performance improvement over existing architectures. This assertion is further tested in the following scenarios.

Scenario 2: Basic simulation with uniform request rate and varying application sizes

Table IV is a collection of the major parameters used in this scenario.

Table IV: Scenario 2 simulation parameters

Parameter		Value
# of servers		300
# of applications		2
App capacity (MIPS)	App1	30000
	App2	5000
Request rate		1 req/sec
Server capacity (MIPS)		40000
Internal variables	RetrieveRate	5x
	RequestRateParam	10
	RetrieveRequestParam	0.2
	ServerProvisioningTime	3 (1.5 sec)
Managers (*sysAC*, *sysVC* & *sysDC*)		PeM
DZConst		1.5

In this scenario, there are 300 servers of 40000 capacity each to be shared amongst two applications (App1 and App2). This means there is a total of initial 12000000 MIPS to share between requests for App1 with 30000 MIPS and App2 with 5000 MIPS. The capacity gap between the two applications is so wide that it may naturally lead to contention with App1 demanding more resources than App2. In this kind of situation where it is easy to underserve one application because of the contention, it is left for the datacentre autonomic managers to decide how best to efficiently allocate resources. Results show that while *sysAC* maintained a proportionate resource allocation (in terms of applications) for the two applications, *sysVC* and *sysDC* prioritised provisioning for App1 with much higher MIPS request. One disadvantage of proportionate provisioning is that it treats requests according to applications (in this case two applications) and not according to capacity (in this case 30000 versus 5000). When this happens, the high capacity application (App1) will be heavily under-provisioned while the low capacity application (App2) will be adequately provisioned (and sometimes over-provisioned) compared to the level of provisioning for App1 as shown in Figure 14 (a) for *sysAC* Offset analysis. Also this amounts to inefficiency and explains why *sysAC* easily exhausts its resources as shown in Table V. Table V shows the results of requests distribution amongst the three managers.

The 'dropped/queued request' analysis shows that in prioritising App1, *sysVC* and *sysDC* dropped more of App2 requests while *sysAC*, which does not drop any application, struggled to cope with the capacity imbalance. For a clearer picture Figure 13 shows how *sysVC* and *sysDC* prioritised App1 over App2.

Table V: High level performance analysis of managers over ten simulation runs of Scenario 2

Sim.	unused server			serviced request			deployed server		
	AC	VC	DC	AC	VC	DC	AC	VC	DC
1	0	118	127	423	242	231	399	227	207
2	0	113	125	465	263	251	422	233	213
3	0	132	145	450	234	225	418	211	191
4	0	120	113	447	248	254	411	211	223
5	0	124	122	440	246	243	405	218	218
6	0	100	120	451	259	250	413	237	221
7	0	108	127	470	265	253	420	239	208
8	0	96	114	434	262	258	404	236	228
9	0	102	116	458	261	257	413	241	222
10	0	107	112	428	250	249	394	225	219
avg	**0**	**112**	**122.1**	**446.6**	**253**	**247.1**	**409.9**	**227.8**	**215**

As can be seen in Figure 13, there is a consistent trend of high rate of dropped App2 requests. This means that more resources were allocated to App1 and thereby starving App2. As this continued, it led to more App2 being dropped as there were limited resources per time to service App2 requests. Also noticeable is the smoothness of provisioning for App1 compared to the bumpiness of provisioning for App2 –this is further explained in the Offset analysis that follows.

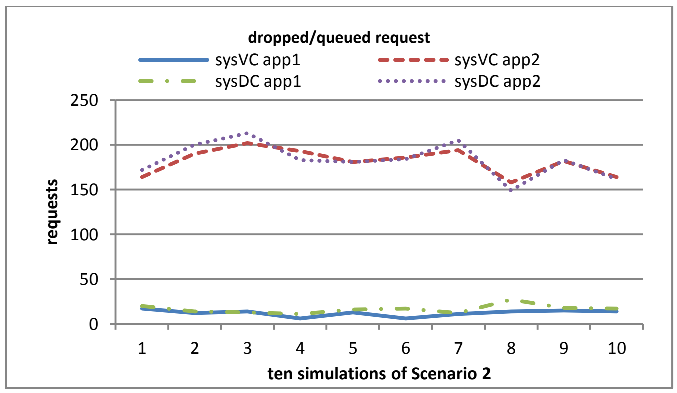
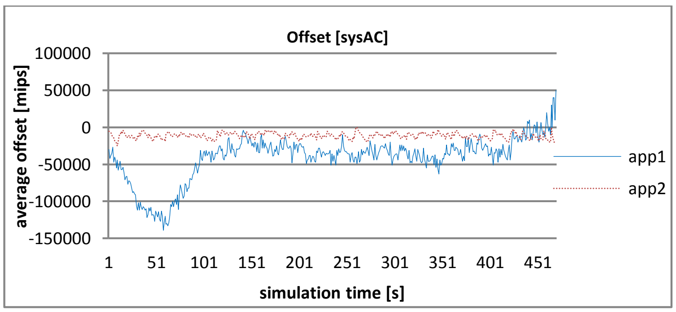


Figure 13: Dropped/queued request analysis for Scenario 2.

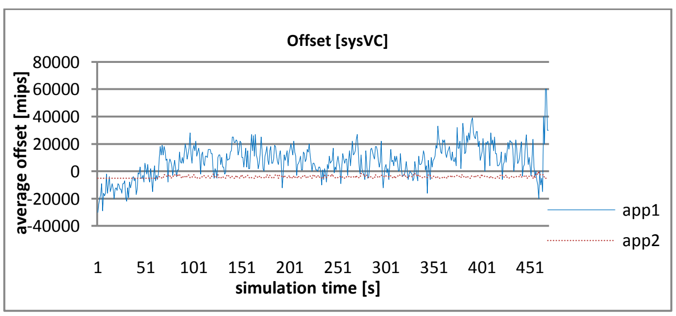
sysAC on the other hand did not drop any request and trying to evenly joggle resources between the highly imbalanced MIPS requests for the two applications meant that more resources per time than necessary are used. This explains why *sysAC* exhausted its resources quite early in the simulation while the other managers have hundreds of servers still unused (Table V). Figure 14 (a) shows that while App2 is about adequately provisioned, App1 is heavily under-provisioned. This is because *sysAC* evenly provisioned for the two applications thereby starving App1, which has very high MIPS requests. So by accepting all requests despite low resource availability *sysAC* under-provisioned for App1 far more than it did for App2 because of the large size of App1 requests. There is no check in *sysAC* to ensure resource availability before requests are accepted.

In Figure 14, App2 offset is maintained at (0 ≥ -18000 MIPS) by *sysAC*, (-1666 ≥ -5000 MIPS) by *sysVC* and (0 ≥ -5000 MIPS) by *sysDC*. Also, App1 offset ranges between (50000 and -139000 MIPS) for *sysAC*, (60000 and -30000 MIPS) for *sysVC* and (30000 and -30000 MIPS) for *sysDC*.

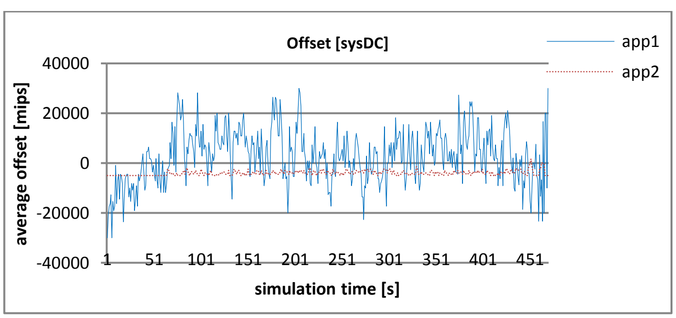
This shows that while *sysAC* treats requests according to applications (i.e., by trying to evenly provision for both applications), *sysVC* and *sysDC* are sensitive to the individual size of requests. As a result, by taking on all requests and attempting an even distribution of resources for both applications, *sysAC* heavily under-provisions for App1 and this also affected its performance for App2. *sysVC* and *sysDC* on the other hand, maintained more balanced resource allocation for both applications in terms of request capacity with *sysDC* showing higher efficiency than *sysVC*. Note that a positive Offset above the optimal provisioning mark amounts to over-provisioning while a negative Offset amounts to under-provisioning. Recall that optimal provisioning mark is defined by the interval ($0 \leq Offset \leq AvgAppCapacity$), which in this case is ($0 \leq$ Offset $\leq ((30000 + 5000)/2)$) –that is, between 0 and 17500 MIPS.



(a) *sysAC* Offset analysis for App1 and App2. App2 is about adequately provisioned (i.e., Offset ≈ 0) while App1 is heavily under-provisioned



(b) *sysVC* Offset analysis for App1 and App2. App2 is about adequately provisioned while App1 over-provisioned (well above the optimal provisioning mark, which is defined by $0 \leq Offset \leq AvgAppCapacity$)



(c) *sysDC* Offset analysis for App1 and App2. App2 is about adequately provisioned while App1 is slightly over-provisioned (slightly above the optimal provisioning mark, which is defined by $0 \leq Offset \leq AvgAppCapacity$)

Figure 14: Individual Offset analysis for scenario 2.

Figure 15 shows the average manager efficiency analysis for all three systems. On the average *sysAC* did not stand up to the complex provisioning condition of Scenario 2 as did the other systems. Figure 15 shows that *sysAC* could not efficiently cope with the level of resource contention experienced between App1 and App2. *sysVC* and *sysDC* show almost the same level of autonomic sophistication however, *sysDC* is shown to be more efficient. Although both systems have the same least under-provisioning value of -17500 MIPS, *sysVC* recorded a maximum over-provisioning value of 27500 MIPS (well above the optimal provisioning mark of 17500) while *sysDC* recorded a maximum positive Offset value of 13500 MIPS (below the optimal provisioning mark). This indicates that *sysDC* is efficiently more sophisticated in handling complex resource allocation scenario that would ordinarily prove difficult for traditional autonomic managers (*sysAC* and *sysVC*) to handle. E.g., this increased efficiency arises from the fact that the *DependabilityCheck* sub-component of *sysDC* enables it to go beyond dropping requests if there are insufficient resources to deploying resources only when it is necessary and efficient to do so.

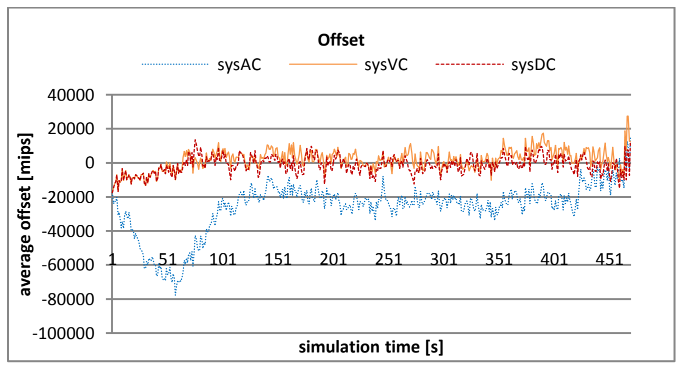


Figure 15: Manager efficiency analysis for scenario 2.

The results analysis of Scenario 2 is a further corroboration of the assertion that the TAArch architecture (represented by *sysDC*) has significant performance improvement over existing architectures. There are two more simulation scenarios to further test this assertion.

Scenario 3: Uniform application size with burst injected at a particular time in the simulation

In this scenario, request rate and application size are kept constant while burst is injected at a particular time (200s) in the simulation. This is similar to Scenario 1 just that a sudden and unexpected disruption is injected into the system. This simulation will measure the robustness of the AMs in adhering to the goal of the system. Another important factor to look at is how long it takes the AMs to recover from the disruption caused by the burst. The impact of the burst is relative to the size of the burst, (which in this case is 2500 ms). Table VI is a collection of major parameters used.

Table VI: Scenario 3 simulation parameters

Parameter		Value
# of servers		300
# of applications		4
Request rate		1 req/sec
Application capacity (MIPS)		20000
Server capacity (MIPS)		40000
Internal variables	RetrieveRate	5x
	RequestRateParam	10
	RetrieveRequestParam	0.2
	BurstSize	2500ms
	ServerProvisioningTime	3 (1.5 sec)
Managers (*sysAC*, *sysVC* & *sysDC*)		PeM
DZConst		1.5

In every simulation, there are 300 servers of 40000 MIPS each. This means there is a total of initial 12000000 MIPS to share between four applications (App1, App2, App3, and App4). Reclaimed servers are subsequently added to this available capacity. The managers receive requests and allocate resources accordingly as long as *AvailableCapacity* is not zero. The reliability of a manager will be measured by its ability to remain efficient under almost all perceivable operating circumstances. Table VII is a number-distribution of requests and services for ten simulation runs of Scenario 3.

Table VII: High level performance analysis of managers over ten simulation runs of Scenario 3

	unused server			**serviced request**			**deployed server**		
	AC	VC	DC	AC	VC	DC	AC	VC	DC
1	0	68	89	453	417	407	306	240	211
2	0	55	74	564	431	418	309	253	230
3	0	61	90	467	430	415	309	248	216
4	0	63	86	481	439	423	307	242	220
5	0	59	79	482	447	431	312	255	232
6	0	57	87	462	426	412	304	246	214
7	0	69	93	444	408	391	307	235	219
8	0	67	94	455	420	404	302	238	209
9	0	63	95	463	424	408	305	248	213
10	0	58	80	453	420	410	304	247	226
avg	**0**	**62**	**86.7**	**472.4**	**426.2**	**411.9**	**306.5**	**245.2**	**219**

On the average, from Table VII, *sysAC* had initiated about 46.2 requests (924000 MIPS) more than *sysVC* and about 60.5 requests (1210000 MIPS) more than *sysDC* but has no extra capacity left to proceed beyond this point. However, *sysVC* and *sysDC* both have about 2480000 MIPS and 3468000 MIPS extra capacity respectively. This means that, under normal circumstances, both systems (*sysVC* and *sysDC*) could conveniently provision for about additional 124 and 173.4 requests respectively. Clearly, *sysDC* is seen to have outperformed the other systems. This is principally because the dead-zone logic of *sysDC* helps it to significantly reduce the number of activated decision boundaries. This means that decisions are not erratically taken, which leads to high efficiency and reliability. Figures 17 – 20 give a breakdown of the performances.

Figure 16 shows how all three managers reacted to the disruption injected at 200s. While *sysVC* and *sysDC* were able to recover after about 9s each (with *sysDC* a bit less than that), it took *sysAC* about 90s to recover. We can also see that *sysDC* reasonably maintained provisioning within the optimal provisioning mark, which in this case is between 0 and 20000 MIPS. There is also a noticeable trend that suggests an extra level of autonomic sophistication in *sysDC* which is also a sign of reliability. Notice that within pre disruption and post disruption recovery both *sysAC* and *sysVC* maintained their level of performances (which nonetheless is averagely about 5000 MIPS above the optimal provisioning mark) while *sysDC*, within the same time frame, switched between two levels of performance as shown by the solid black line. This is the effect of dynamic (re)tuning of the *DZWidth* by *sysDC*. This capability enables *sysDC* to systematically track the system's goal (in this case maintaining reliability and efficiency within the optimal provisioning mark) by dynamically retuning its decision boundary. As shown in Figure 16, before the disruption *sysDC* maintained a steady and continuous level of efficiency by keeping *DZConst* at 1.5 but as soon as the disruption sets in it quickly retunes itself and reduced the *DZConst* to 1. At this point the manager stopped accepting further requests (as the datacentre is now receiving torrential streams of requests) but the initial shock (caused by the lag between when the disruption started and when the manager shuts its door) meant that a few resources were released to mitigate the effect of the situation. This will instantly start pushing up *Offset* until the datacentre normalises and then as shown *sysDC* retunes its decision boundary by returning *DZConst* back to 1.5. So while *sysAC* is heavily affected by a disruption of this magnitude and *sysVC* shows a remarkable level of robustness, *sysDC* shows a longer term ability to sensitively throttle its behaviour to efficiently and reliably track the goal of the entire system.

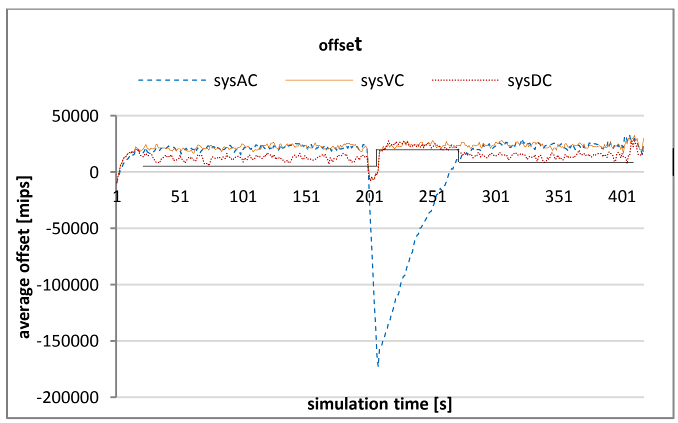


Figure 16: Manager efficiency analysis for scenario 3. The black solid line indicates sysDC's dynamic tuning of dead-zone boundary. The manager started with a *DZConst* of 1.5 (left lower part of the line) then changed to *DZConst* of 1 (high part) and then back to *DZConst* of 1.5.

Figure 17 shows that while *sysVC* and *sysDC* responded to the disruption by rejecting requests as soon as they were overwhelmed thereby pushing down their server deployment rate, *sysAC* responded by deploying even more servers to meet the current service demand. Despite deploying more servers *sysAC* still could not meet up with demand rate, which ultimately affected its SLA achievement (Figure 18). This is because the provisioning rate, (which is dependent on *ProvioningTime*) could not keep up with the rate at which servers are deployed. As a result *sysAC* had more servers (almost tripling that of *sysDC*) overshooting their *ProvisioningTime* thereby getting redundant and pushing up delay cost as well.

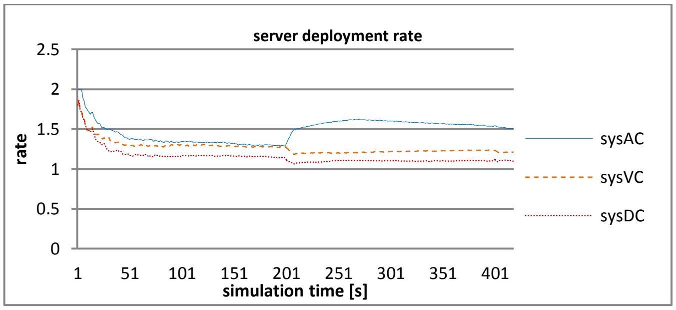


Figure 17: Server deployment rate analysis for scenario 3.

As the datacentre settles (after the disruption) *sysAC* starts normalising the rate of server deployment but because there is already a huge backlog of requests (about 173000 MIPS as in Figure 16) it takes *sysAC* a long time to recover. This also contributes to why it quickly exhausts its resources. *sysVC* and *sysDC* on the other hand, with a small backlog of about 7500 MIPS, need not deploy more resources than the ordinary (Figure 17) but gradually absolves the backlog allowing them to quickly recover.

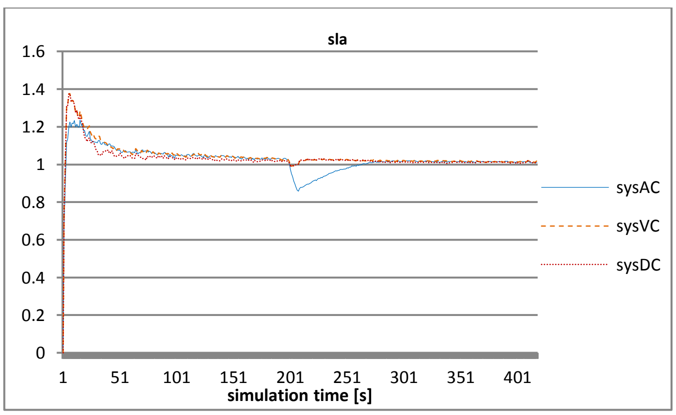


Figure 18: Service level achievement (SLA) analysis for scenario 3.

High level of deployment rate (inefficient deployment of more MIPS than necessary) also leads to high cost (in terms of excess MIPS) of servicing individual requests. This means that the rate at which servers enter the provisioning queue is much higher than the rate they leave the queue. The rate for *sysAC* almost doubles that of *sysVC* and almost triples that of *sysDC*. This leads to increasing number of redundant servers in the queue, which contributes to delay cost.

The results analysis of Scenario 3 shows that it is absolutely inefficient and unreliable to run a datacentre with a manager based on *sysAC*. While *sysVC* based AMs are more robust, their robustness is limited in terms of the extent of sensitivity to system's goal under unfamiliar circumstances in which *sysDC* based AMs are more sophisticated and dynamically reliable. This further corroborates the assertion that the TAArch architecture (*sysDC*) has significant performance improvement over existing architectures.

Scenario 4: Varying application sizes with inconsistent request rate

This is the most complex scenario with a combined effect of Scenarios 2 and 3 put together. The complexity presented by this scenario (i.e., a combined effect of resource contention and two injected disruptions) allows us to further test the robustness of these systems by stretching their capabilities to extremes. Table VIII is a collection of the major parameters used in this scenario. As in previous scenarios, results presented are based on average of ten different simulation runs.

Table VIII: Scenario 4 simulation parameters

Parameter		Value
# of servers		400
# of applications		2
App capacity (MIPS)	App1	30000
	App2	15000
Request rate (initial)		1 req/sec
Server capacity (MIPS)		40000
Internal variables	RetrieveRate	5x
	RequestRateParam	10
	RetrieveRequestParam	0.2
	BurstSize	1500ms
	ServerProvisioningTime	3 (1.5 sec)
Managers (*sysAC*, *sysVC* & *sysDC*)		PeM
DZConst (initial)		1.5

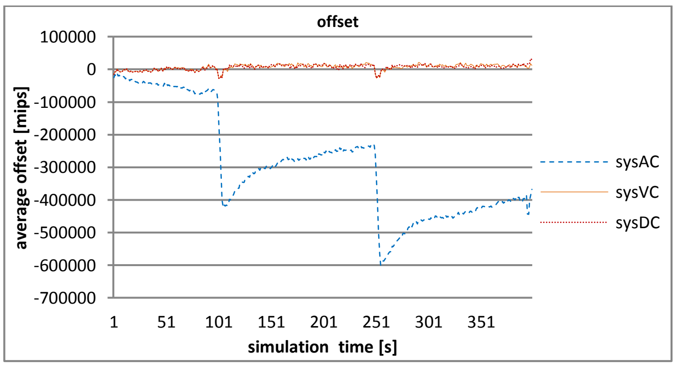
In every simulation of this scenario, there are 400 servers of 40000 MIPS each to be shared amongst two applications (App1 and App2). This means there is a total of initial 16000000 MIPS to share between requests for App1 with 30000 MIPS and App2 with 15000 MIPS. Table IX is a number distribution of requests and services for ten simulation runs of Scenario 4.

Table IX: High level performance analysis of managers over ten simulation runs of Scenario 4

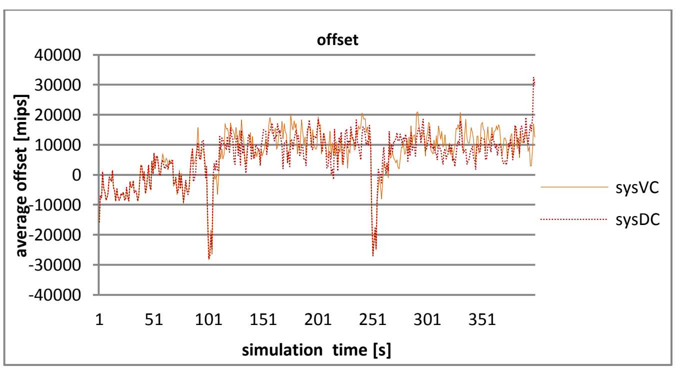
	unused server			serviced request			deployed server		
	AC	VC	DC	AC	VC	DC	AC	VC	DC
1	0	109	120	474	395	394	435	339	316
2	0	124	133	465	387	382	433	325	303
3	0	123	125	471	400	397	443	330	314
4	0	112	114	473	395	400	439	343	321
5	0	114	130	476	398	402	440	335	304
6	0	118	124	473	393	398	439	331	308
7	0	115	117	468	393	394	437	336	320
8	0	113	122	468	398	396	435	330	307
9	0	113	116	476	395	401	444	342	322
10	0	110	115	476	398	394	446	337	323
avg	**0**	**115**	**122**	**472**	**395**	**393**	**439**	**335**	**314**

Results reveal that *sysAC* is not adequately robust in such complex situations as in Scenario 4. The system is heavily inefficient in handling this type of situation (Figure 19 (a)). Its algorithm, which maintains proportionate provisioning with respect to number of applications as against capacity of requests, was disorientated by the level of contention and disruption experienced.

As shown in Figure 19 the first burst was injected at 100s while the second was injected at 250s. *sysAC* is limited in its ability to handle complex situations and so cannot be relied upon to operate large scale and complex datacentres. *sysVC* and *sysDC* both have a wide range of operability in complex situations. However, a closer look at *sysVC* and *sysDC* in this scenario reveals a unique change in expected (as observed in previous results) trend. The highlighted bits of Table IX show that *sysDC* dropped fewer requests than *sysVC* and thereby initiating more requests. Under normal circumstances, as observed in previous scenarios, *sysVC* usually would drop fewer requests than *sysDC*. In this situation the level of disturbance (as a result of resource contention and erratic request disorder) in the datacentre led to instability in *sysVC*, which caused it to over react by inefficiently dropping requests. This instability reveals a weakness in design because in real-life datacentres such disturbances (like sudden request spikes) do occur and managers are expected to adequately stabilise the entire system under such circumstances. *sysDC* on the other hand, with the capability of a longer term view of the entire system, was able to take on more requests.



(a) Manager efficiency analysis of all three systems



(b) Manager efficiency analysis for *sysVC* and *sysDC*.

Figure 19: Manager efficiency analysis for Scenario 4. Bursts affect all managers at 100s and 250s time frames

However, this achievement is with associated tradeoff in delay cost (Figure 20). This shows that *sysDC* is more sensitive to the relationship between requested MIPS and available MIPS. For example, in a situation where *sysVC* dropped a number of requests following a fixed decision boundary (when there is lack of immediate available resources to handle incoming requests), *sysDC* used a dynamic decision boundary to accommodate more requests allowing it to efficiently use up its available resources. By taking on more requests, *sysDC* trades off delay cost, which is not so much of importance but at the same time improves scheduling efficiency, which is of more importance. Interestingly, the efficiency level is not affected –Figure 19 (b) shows that there is no significant difference in efficiency performance of both *sysVC* and *sysDC*. So what we have is a situation where, on the average, *sysDC* utilised significantly fewer resources (313.8 : 334.8 servers) to serve slightly higher amount of requests (395.8 : 395.2 requests) as *sysVC* (Table IX) resulting in improved efficiency (Figure 19 (b)) for *sysDC* and approximately same level of SLA (Figure 21) and delay cost (Figure 20) achievement for both *sysVC* and *sysDC*.

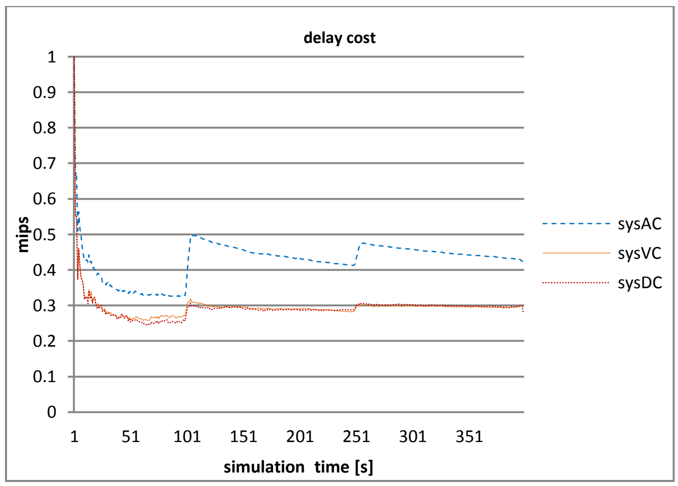


Figure 20: Cost analysis for Scenario 4.

There is consistent corroboration of the fact that *sysAC* is limited in the range of its operational scope when it comes to complex situations. Scenario 4 results show that it is highly expensive, inefficient and unreliable to operate complex datacentres with autonomic managers based on *sysAC*. However, *sysAC* based managers may suffice for simple and basic datacentres. On the other hand, *sysDC* has shown consistent reliability in all tested scenarios. The level of robustness exhibited in this scenario by *sysDC* is a clear indication that it is not a hard-wired one-directional self-managing system. For example, in this scenario we have seen that *sysDC* does not only act when *sysVC* is taking more actions than necessary but also when it is taking fewer actions than necessary. So it can be said that *sysDC* is capable of reducing inefficient adaptation (e.g., when *sysVC*'s decisions are erratic) as well as increasing adaptation when it is necessary and efficient to (e.g., when *sysVC* is not making decisions frequently enough). This capability of increased adaptation is shown in Table IX and illustrated in Figures 20 to 22 –*sysDC* is able to maximise resources while achieving the same level of performance as *sysVC*.

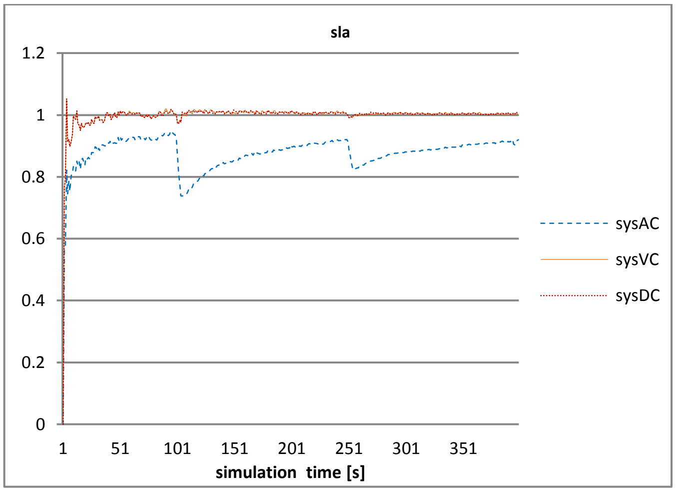


Figure 21: Service level achievement (SLA) analysis for scenario 4.

From the results of the four experimental scenarios presented above we can conclude that *sysAC* has a narrow envelope of operational conditions in which it is both self-managing and returns satisfactory behaviour. On the other hand, *sysVC* tends towards a wider operational envelope with increased efficiency and satisfactory behaviour, but once the limits of that envelope are reached the efficiency and reliability of the system drops. In moderate operational complexities *sysVC* performs adequately efficient but fluctuates rapidly and may need human input to override some decisions that lead to instability in the case of highly erratic and complex situation, which for example *sysDC* can deal with autonomically. Results have shown that *sysDC* is sufficiently sophisticated to operate efficiently and yield satisfactory results under almost all perceivable operating circumstances. So we can now confidently conclude that the proposed trustworthy autonomic architecture (represented by *sysDC*) has significant performance improvement over existing architectures and can be relied upon to operate (or manage) almost all level of datacentre scale and complexity.

Generally, the combination of DC and VC (VC + DC) leads to significant performance improvement over VC. However, the extent of this improvement is application and context dependent. Results show that there are circumstances in which performance improvement is evident from VC + DC as well as circumstances in which improvement is not evident. Complex applications with the possibility of unexpected behaviour patterns, e.g., large scale datacentres with complex algorithms, will usually experience improvement with VC + DC. Also, applications that are sensitive to fluctuating environmental inputs (i.e., depend on volatile environmental information for decision-making), for example, auto stock trading systems are expected to see greater benefit from VC + DC. On the other hand, there are applications that are not expected to see any benefit. Example includes small scale datacentres with predefined request rate and request capacity.

V. CONCLUSION

This paper has presented a new trustworthy autonomic architecture (TAArch). Different from the traditional autonomic solutions, TAArch consists of inbuilt mechanisms and instrumentation to support run-time self-validation and trustworthiness. The architecture guarantees self-monitoring over short time and longer time frames. At the core of the architecture are three components, the *AutonomicController*, *ValidationCheck* and *DependabilityCheck*, which allow developers to specify controls and processes to improve system trustability. We have presented a case example scenario to demonstrate the workings of the proposed approach. The empirical analysis case example scenario is an implementation of a datacentre resource request and allocation management designed to analyse the performance of the proposed TAArch architecture over existing autonomic architectures. Results show that TAArch is sufficiently sophisticated to operate efficiently and yield satisfactory results under almost all perceivable operating circumstances. Analyses also show that the proposed architecture achieves over 42% performance improvement (in terms of reliability) in a complex operating circumstance. It is also safe to conclude that the proposed trustworthy autonomic architecture has significant performance improvement over existing architectures and can be relied upon to operate (or manage) almost all level of datacentre scale and complexity.

The importance of trustworthiness in computing, in general, has been echoed in the Computing Research Association's 'four grand challenges in trustworthy computing' [31] and Microsoft's white paper on Trustworthy Computing (TC) [32]. The Committee on Information Systems Trustworthiness in [33] defines a trustworthy system as one which does what people expect it to do – and nothing more – despite any form of disruption. Although this definition has been the driving force for achieving trustworthiness both in autonomic and non-autonomic systems, the peculiarity of context dynamism in autonomic computing places unique and different challenges on trustworthiness for autonomic systems. Validation for example, which is an essential requirement for trustworthiness, can be design-time based for non-autonomic systems but must be run-time based for autonomic systems. Despite the different challenges, it is generally accepted that trustworthiness is a non-negotiable priority for computing systems. For autonomic systems, the primary concern is not how a system operates to achieve a result but how dependable is that result from the user's perspective. For complete reliance on autonomic computing systems, the human user will need a level of trust and confidence that these systems will satisfy specified requirements and will not fail. It is not sufficient that systems are performing within requirement boundaries, outputs must also be seen to be reliable and dependable. This is necessary for self-managing systems in order to mitigate the threat of losing control and confidence [34]. We posit that such capabilities need to be built in as integral part of the autonomic architecture and not treated as add-ons.

The traditional MAPE-based autonomic architecture as originally presented in [2] has been widely accepted and autonomic research efforts are predominantly based on this architecture's control loop. We must admit that a good research success has been achieved using the MAPE-based architecture. However, we suppose, like others, e.g., [7][10], that this architecture is vague and thus cannot lead to the full goal of autonomic computing. For example, the MAPE-based architecture does not explicitly and integrally support run-time self-validation, which is a prerequisite for trustworthiness.

REFERENCES

[1] T. Eze, R. Anthony, C. Walshaw, and A. Soper, "A New Architecture for Trustworthy Autonomic Systems," in *Proceedings of the Fourth International Conference on Emerging Network Intelligence: (EMERGING) 2012*, pp. 62-68 Barcelona, Spain.

[2] IBM Autonomic Computing White Paper, An architectural blueprint for autonomic computing, 3rd edition, June 2005. Available via http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf last viewed 18th December 2013.

[3] M. Huebscher and J. McCann, "A survey of autonomic computing—degrees, models, and applications," *ACM Computer Survey (CSUR)*, Volume 40, Issue 3, August 2008, Article 7.

[4] C. Reich, K. Bubendorfer, and R. Buyya, "An autonomic peer-to-peer architecture for hosting stateful web services", in *Proceedings of the Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, pp. 250-257, 2008.

[5] P. de Grandis and G. Valetto, "Elicitation and utilization of utility functions for the self-assessment of autonomic applications," *in Proceedings of the sixth International Conference on Autonomic Computing (ICAC), 2009*, Barcelona, Spain.

[6] C. Dorn, D. Schall, and S. Dustdar, "A model and algorithm for self-adaptation in service-oriented systems," in *Proceedings of the seventh IEEE European Conference on Web Services* (ECOWS), 2009, pp. 161 – 170, Eindhoven, The Netherlands.

[7] H. Shuaib, R. Anthony, and M. Pelc, "A Framework for Certifying Autonomic Computing Systems", in *Proceedings of the Seventh International Conference on Autonomic and Autonomous Systems: (ICAS) 2011*, Venice, Italy.

[8] T. Eze, R. Anthony, C. Walshaw, and A. Soper, "Autonomic Computing in the First Decade: Trends and Direction," in *Proceedings of the Eighth International Conference on Autonomic and Autonomous Systems (ICAS)* 2012, St. Maarten, Netherlands Antilles.

[9] J. Kephart and D. Chess, "The Vision of Autonomic Computing," Computer, vol. 36, no. 1, pp. 41-50, 2003.

[10] A. Diniz, V. Torres, and C. José, "A Self-adaptive Process that Incorporates a Self-test Activity," *Monografias em Ciência da Computação*, number 32/09, 2009, Rio – Brazil.

[11] H. Chan, A. Segal, B. Arnold, and I. Whalley, "How Can We Trust an Autonomic System to Make the Best Decision?" in *Proceedings of the second International Conference on Autonomic Computing (ICAC), 2005*, Seattle, USA.

[12] J. Hall and L. Rapanotti, "Assurance-driven design in Problem Oriented Engineering," in *International Journal On Advances in Intelligent Systems (IntSys)*, volume 2, number 1, 2009, pp. 26-37.

[13] S. Kikuchi, S. Tsuchiya, M. Adachi, and T. Katsuyama, "Constraint Verification for Concurrent System Management Workflows Sharing Resources," in *Proceedings of the third International Conference on Autonomic and Autonomous Systems (ICAS), 2007*, Athens, Greece.

[14] X. Li, H. Kang, P. Harrington, and J. Thomas, "Autonomic and trusted computing paradigms," in *Lecture Notes in Computer Science*, Volume 4158, 2006, pp. 143-152.

[15] S. Anderson, M. Hartswood, R. Procter, M. Rouncefield, R. Slack, J. Soutter, and A. Voss, "Making Autonomic Computing Systems Accountable," in *Proceedings of the fourteenth International Workshop on Database and Expert Systems Applications (DEXA), 2003*

[16] R. Anthony, "Policy-based autonomic computing with integral support for self-stabilisation," *International Journal of Autonomic Computing*, Vol. 1, No. 1, 2009, pp. 1–33.

[17] R. Anthony, "Policy-centric Integration and Dynamic Composition of Autonomic Computing Techniques," in *Proceedings of the fourth International Conference on Autonomic Computing (ICAC), 2007*, Florida, USA.

[18] J. Heo and T. Abdelzaher, "AdaptGuard: Guarding Adaptive Systems from Instability," in *Proceedings of the sixth International Conference on Autonomic Computing (ICAC), 2009*, Barcelona, Spain.

[19] J. Hawthorne, R. Anthony, and M. Petridis, "Improving the Development Process for Teleo-Reactive Programming Through Advanced Composition," in *Proceedings of the Seventh International Conference on Autonomic and Autonomous Systems (ICAS) 2011*, Venice, Italy.

[20] D. Richards, M. Taylor, and P. Busch, "Expertise Recommendation: A triangulated approach," in *International Journal On Advances in Intelligent Systems (IntSys)*, volume 2, number 1, 2009, pp. 12-25.

[21] T. King, A. Ramirez, R. Cruz, and P. Clarke, "An Integrated Self-Testing Framework for Autonomic Computing Systems," *Journal of computers*, vol. 2, no. 9, 2007, pp. 37-49.

[22] T. Eze, R. Anthony, C Walshaw, and A. Soper, "A Methodology for Evaluating Complex Interactions between Multiple Autonomic Managers", in *Proceedings of the Ninth International Conference on Autonomic and Autonomous Systems* (ICAS), 2013, Lisbon, Portugal.

[23] H. Li and S. Venugopal, "Using Reinforcement Learning for Controlling an Elastic Web Application Hosting Platform," in *Proceedings of the eighth International Conference on Autonomic Computing (ICAC), 2011*, Karlsruhe, Germany.

[24] T. Yu, R. Lai, M. Lin and B. Kao, "A Fuzzy Constraint-Directed Autonomous Learning to Support Agent Negotiation," in *Proceedings of the third International Conference on Autonomic and Autonomous Systems (ICAS), 2007*, Athens, Greece.

[25] R. Das, J. Kephart, J. Lenchner, and H. Hamann, "Utility-function-driven energy-efficient cooling in data centers," *in Proceeding of the Seventh International Conference on Autonomic Computing (ICAC)*, 2010, New York, USA.

[26] I. Goiri, J. Fit´o, F. Juli`a, R. Nou, J. Berral, J. Guitart and J. Torres, "Multifaceted Resource Management for Dealing with Heterogeneous Workloads in Virtualized Data Centers*," in Proceedings of Eleventh IEEE/ACM*

International Conference on Grid Computing (GRID), 2010, Brussels, Belgium.

[27] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle, "Managing Energy and Server Resources in Hosting Centers," ACM SIGOPS Operating Systems Review, vol. 35, no. 5, pp. 103–116, 2001.

[28] J. Berral, R. Gavalda, and J. Torres, ""Living in Barcelona" Li-BCN Workload 2010," *Technical Report* LiBCN10, 2010, Barcelona Supercomputing Centre, Barcelona, Spain.

[29] M. Pretorius, M. Ghassemian, and C. Ierotheou, "An investigation into energy efficiency of data centre virtualisation," *in Proceedings of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 2010, Fukuoka, Japan.

[30] M. Pretorius, M. Ghassemian, and C. Ierotheou, "Virtualisation –Securing a Greener Tomorrow with Yesteryear's Technology," *in Proceeding of the Twelfth IFIP/IEEE International Symposium on Integrated Network Management* (IM 2011), 2011, Dublin, Ireland.

[31] Computing Research Association, "Four Grand Challenges in Trustworthy Computing," in *Proceedings of Second Conferences on Grand Research Challenges in Computer Science and Engineering*, November 16–19, 2003.

[32] C. Mundie, P. Vries, P. Haynes, and M. Corwine, "Trustworthy Computing: Microsoft White Paper," *Microsoft Corporation*, October 2002.

[33] F. Schneider, "Trust in Cyberspace," *Committee on Information Systems Trustworthiness, National Academy Press*, 1998, Washington, D.C.

[34] L. Yang and J. Ma, Introduction to the Journal of Autonomic and Trusted Computing. American Scientific Publishers www.aspbs.com/joatc.html 26/08/13 last viewed 28/05/2014

APPENDIX A: TAArch Application

The simulations of this paper are performed using the TAArch Application. To understand the workings of the application let us consider Figure A, which is a screen shot of a basic resource allocation simulation with 75 servers (x) and 4 applications (ix). The user selects the number of servers and applications and this will populate the S_i and A_j pools respectively (labels x and ix). The application supports two experiments ('Normal Simulation' and 'Interoperability', which is not covered here) as shown (iii) and in this case the 'Normal Simulation' option is selected, which will automatically check the *PeM* autonomic manager option (vi). Then the actual manager is selected, which in this case is the [AC+VC+DC] option representing all three managers. As shown (vii) the DZWidth can be manually controlled by the user or dynamically tuned by the system depending on which option is selected. Before the simulation starts it is possible to set the internal variables through (xiv) to user preferences. The possibility of changing the internal variables is deactivated (as shown by xiv) as soon as the simulation starts. Change of server capacity is also deactivated (i) as soon as simulation starts. Meanwhile, application size (i), which is an external variable, can be changed at any time in the simulation. Once all parameters are set the simulation can be started by clicking 'Run Simulation'. For the purpose of this example the shutdown server pool $\check{S}_i$ is not used (xi) –it is only used for the 'Interoperability' simulation.

Once the simulation starts, the manager starts populating the Ų pool (xiii). The view of this pool shows current and live updates of process status. 'Available capacity' shows running capacity available to serve individual application request while 'Run'g requests' are the total running individual request capacity. 'Offset' is the difference between running request capacity and available capacity. 'Server_ID' shows the collection of servers currently providing services for individual application request. Depending on the number of servers in use, some of the allocated servers may no longer be visible in the Ų pool but can be viewed from the respective individual pool (xii). The provisioning servers, that is, servers that are been configured in the queue can be viewed through (ii). Individual results for the managers are displayed in (iv) and (v). Also as stated, data displayed below (viii) and in (ii) are for AC (*sysAC*). Although there is provision for live graphing of results through the 'Show Graph' button, complete result values can be exported to Excel Sheet through the 'Export Results' button (vii).

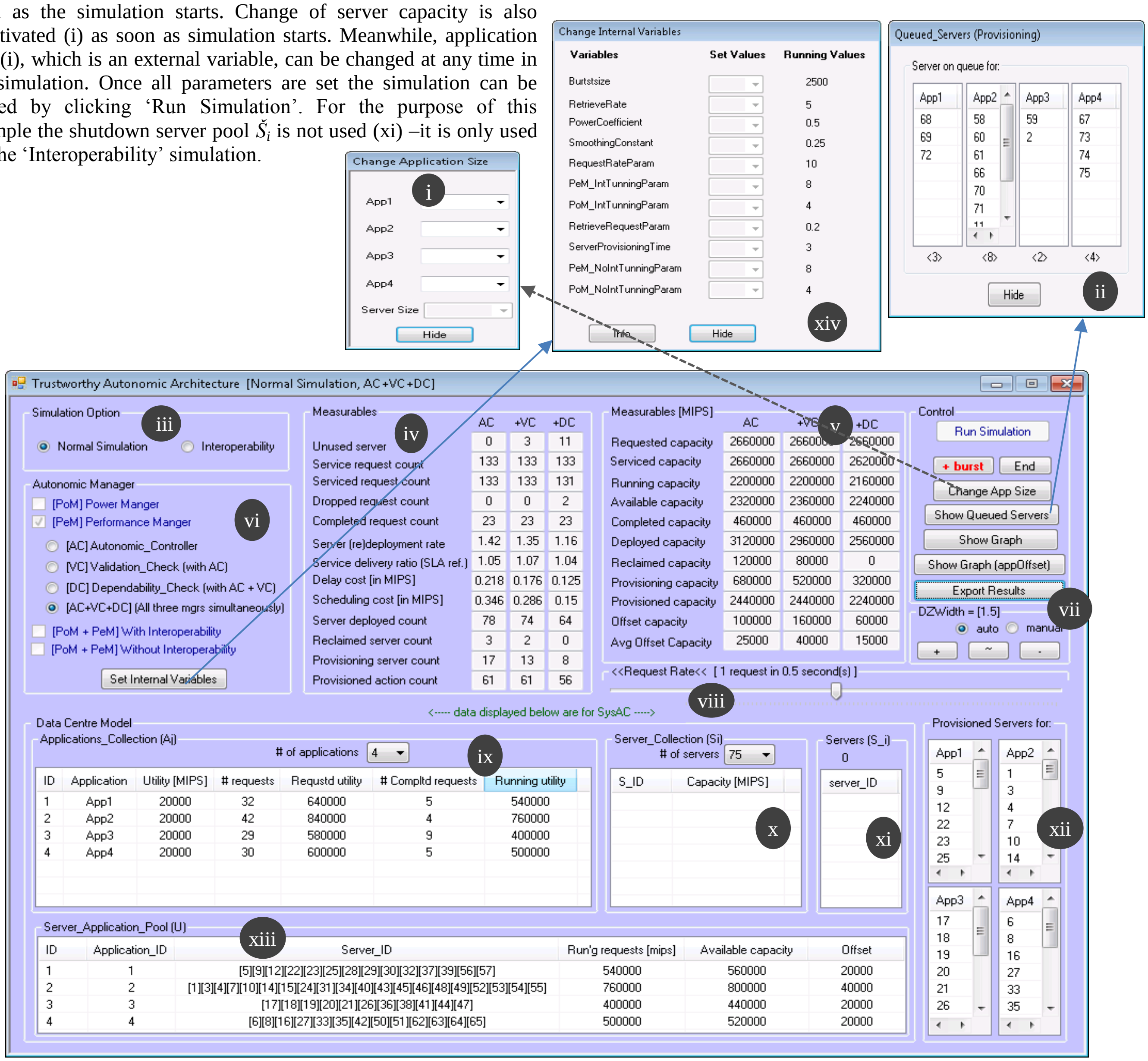


Figure A: Simulation screen shot showing TAArch application front end.

Creation of Adaptive User Interfaces Through Reconfiguration of User Interface Models Using an Algorithmic Rule Generation Approach

Benjamin Weyers
RWTH Aachen University
Virtual Reality Group
JARA-High Performance Computing
Seffenter Weg 23, 52074 Aachen, Germany
Email: weyers@vr.rwth-aachen.de

Nikolaj Borisov and Wolfram Luther
University of Duisburg-Essen
Department of Computer Science and
Applied Cognitive Science
Lotharstr. 63, 47057 Duisburg, Germany
Email: nikolaj.borisov@uni-due.de, luther@inf.uni-due.de

***Abstract*—Inefficient and error-prone interaction between human operators and technical systems was the reason for various catastrophic accidents in the past. User interfaces implement the communication between a human user and a technical system which is the reason why inaccurate design of user interfaces has been identified as one major factor for those errors. The use of adaptive user interfaces is one possible solution to reduce inefficient interaction by adapting the user interface to a specific user, task, or context. However, currently no self-contained formal approach exists that allows for the creation of adaptive user interfaces despite various advantages of formal methods: interaction becomes verifiable, formal methods close the gap between modeling and implementation by using executable formal languages, and they allow for using existing rewriting concepts making formal models adaptable. This paper introduces a new approach to a formal rule generation concept, which enables a flexible creation of adaptive user interfaces. This concept is based on a formal modeling and reconfiguration approach for the creation and adaptation of user interfaces. The applicability of this approach will be shown through an implementation of an adaptive user interface for adaptive automation. The main contribution of the presented work is a new concept for rule generation that is capable of adapting formally modeled user interfaces.**

Keywords–Formal Modeling; User Interface; Adaptive User Interface; Formal Reconfiguration; Rule Generation.

I. Introduction

Inefficient and error-prone interaction occurs during the use of interactive systems if the user interfaces are not sufficiently developed with respect to the needs and abilities of the user or user group [1]. As past events have shown, these errors can lead to catastrophic accidents, such as the disaster in Chernobyl [2]. Adaptive as well as adaptable user interfaces are primarily developed in order to increase human performance by changing the user interface according to a specific user, task, environment, context, or situation [3]. These kinds of user interfaces have shown a high potential in increasing usability [4], reducing errors in interaction [5], or in simplifying interaction with complex systems [6]. Furthermore, formal modeling approaches for user interfaces offer various advantages, like making it possible to directly execute or verify created models. Nevertheless, to our knowledge a self-contained and flexible formal approach for the implementation of adaptive user interfaces has not yet been discussed. Therefore, this work presents a formal modeling and reconfiguration concept for the creation of user interfaces that is extended by an algorithmic rule generation approach as a first step towards a generic and formal generation of adaptive (and adaptable) user interfaces.

The main difference between adaptive and adaptable user interfaces is the corresponding instance applying an adaptation to the user interface. An adaptable user interface is mainly changed manually by the user using tools. In contrast, adaptive user interfaces are changed by a technical implementation. For adaptive user interfaces, the type of adaptation is usually defined in an algorithmic fashion that is further parameterized by data selected from various sources, the interaction between user and the system, or from the system itself. Therefore the implementation of an adaptive user interface involves the type of user interface description or implementation on the one hand, and the adaptation concept that changes the user interface implementation and thereby influences the interaction, on the other. The description of a user interface can be divided into two parts: the *physical representation* and the *interaction logic* [7]. The physical representation covers all elements that are directly accessible by the user. In case of a classic graphical user interface, these elements can be buttons, sliders, or text fields. arranged in a specific layout. The interaction logic specifies the data-based communication between physical representation and the system to be controlled, as well as the logic and data-based dependencies between elements of the physical representation. Thus, interaction logic can also be denoted as the behavioral model of a user interface.

Based on this differentiation between the outward appearance of a user interface and its behavior, the adaptation (whether adaptable or adaptive) can be applied either on the physical representation, the interaction logic, or both. Using this concept combined with a set of tools for implementing an adaptable user interface, we conducted various evaluation studies. For instance in [5], we discussed a study that showed a significant error reduction in controlling a complex technical system by user-side adaptation of the physical representation and the interaction logic applied to an initial user interface model. The whole concept has been implemented based on a formal modeling approach for user interfaces, which is based on this two-layered model. To this end, the physical representation has been formalized using an XML-based description while interaction logic has been formalized using a graph-based approach using reference nets, a variant of Petri nets. Furthermore, this has been combined with a formal rule-based reconfiguration concept, which is controlled through an interactive system by the user.

Apart from these aspects, formal modeling approaches offer further advantages to the modeling and reconfiguration of user interfaces. A formal user interface model can be directly executed using a simulator or interpreter. The combination of this approach with a rule-based reconfiguration technique creates a self-contained formal modeling approach. This supports the generation of adaptable and adaptive user interfaces, because a generic (formal) reconfiguration system can either be controlled by the user through an interactive system (as shortly described above) or by a system that generates rules algorithmically. This close integration of model-based creation and reconfiguration enables adaptation of user interfaces without losing the focus of creating computer-based systems [8]. This is a first step towards closing the gap between modeling and implementation of interactive systems [9]. Finally, this self contained approach of user interface modeling and reconfiguration prevents loss of information that can occur if a formal model is adapted by some informal or non-deterministic concept.

To extend this basic approach of formal adaptable user interfaces to the creation of adaptive user interfaces, the rule generation process necessary for the adaptation of user interface models has to be extended to enable computer-based and generic generation of reconfiguration rules. Therefore, the main concept introduced in this paper combines a description language for defining rule classes accompanied with a set of algorithms, which enable a software tool to instantiate a rule class based on a set of input data. A rule class further specifies a rule skeleton describing a basic structure of a rule that is to be instantiated. According to data defining and influencing the adaptation of a user interface, various channels can be identified, such as sensory data, a user model, or data generated by the controlled process (as discussed above). For instance, sensory data can be gathered to represent the context in which the interaction takes place. Nevertheless, various other types of data and data sources can be identified, which cannot be discussed completely in this paper. The main reason for this is that provided data is highly use case dependent. For instance, adaptivity of user interfaces can also introduce the user into the adaptation loop, such that she provides data or triggers the adaptation. Therefore, the presented work does not specify the type of data source but will support the description of various data types using a formal type specification language, called Resource Description Framework (RDF) [10]. This makes the definition (language) of rule classes independent from a specific use case by abstracting from the explicit data source to a data type that has to be delivered to the rule generation algorithm during runtime. Thus, the whole adaptation process becomes "semi-automatic" through the option of introducing the user into the loop.

Beside defining input data necessary for the instantiation of a rule class, various algorithms are discussed in this work offering functionality to the instantiation of rule classes. The aforementioned rule skeletons are defined based on grammars using nonterminal symbols for graphs and graph inscriptions. Thus, on the one hand, matching of nonterminal symbols in graphs and in inscriptions has to be performed based on the user interface model to be adapted, as well as on the input data as specified in the class and provided during runtime. On the other hand, algorithms for traversing a given graph are discussed regarding the extraction of certain parts in the user interface model that are part of its reconfiguration. Finally, changes of the visible part of the user interface have to be defined in the class and have to be finally applied to the user interface's physical representation. In conclusion, the main contribution of this work is an algorithmic approach for creating adaptive user interfaces based on a newly developed rule generation concept that defines the adaptation logic of such user interface models.

Before defining the semi-automatic rule generation, related work will be discussed in Section II identifying previous work done in the context of adaptive user interfaces, formal user interface modeling, reconfiguration, and adaptation. Section III describes a process to formal modeling and reconfiguration of user interfaces, which is executable and offers mechanisms for model-intrinsic adaptation through graph transformation systems. As computer-based adaptation of user interfaces assumes the accessibility of context information in various senses in a system's architecture as a formal model or description. Section IV presents a modeling approach of rule classes and an algorithmic rule generation concept that makes system-side adaptation of formal user interface models possible. In Section V, the whole adaptation process will be applied to the use case of adaptive automation and will show how the approach can be used in automated system control. Finally, Section VI will conclude the paper and will discuss future work aspects. The work at hand extends the previous paper by Weyers presented on the IARIA CENTRIC workshop 2013 [1]. Please consider that certain parts (Section V-B in particular) of this article have been reused in the work at hand to underline its origin.

II. Related Work and State of the Art

Adaptive user interfaces are nowadays an integral part of human-computer interaction research. Various works can be identified discussing different usecase dependent views to adaptive user interfaces, which have a similar goal: making interaction between a user and a system less error-prone and more efficient. Jameson [11] gives a broad overview of various functions of adaptive user interface that support this goal. One function he identifies is denoted as "supporting system use". He splits this category further up into the functions of "taking over parts of routine tasks", "adapting the interface", and "controlling a dialog", which are of main interest in the context of this work. Lavie et al. [12] identify certain dimensions of what data or knowledge is needed for the implementation of adaptive user interfaces: the task, the user, and the type of situation in which the interaction takes place. The latter, they characterize as routine vs. non-routine situations. Finally, they discuss the level of adaptivity that specifies the amount of adaptation applied to the user interface. These functions are provided by various implementations and work that has been done on adaptive user interfaces. A general overview of task and user modeling is given by Langley [13] and Fischer [14].

Nevertheless, this work does not focus on how the data and knowledge is gathered or described, but concentrates on how this data can be used for applying changes to a given user interface model.

However, various examples of the successful implementation of adaptive user interfaces can be found, which consider the former discussed aspects. For instance, Reinecke and Bernstein [4] describe an adaptive user interface implementation that takes cultural differences of users into consideration. They showed that users were 22% faster using this implementation. Furthermore, they made fewer errors and rated the adapted user interface as significantly easier to use. Cheng and Liu [15] discuss an adaptive user interface using eye-tracking data to retrieve user's preferences. Kahl et al. [16] present a system called SmartCart, which provides a technical solution for supporting a customer during her shopping process. It is able to provide context-dependent information and support, such as a personalized shopping list or a navigation service. Furthermore, in the context of ambient intelligent environments, Hervas and Bravo [17] present their approach of adaptive user interfaces, which is based on Semantic Web technologies. The so called ViMos framework is able to generate visualization services for context-dependent information.

All presented approaches and implementation have in common that they do not support a full-fledged formal modeling approach for the adaptation of user interfaces. Still, formal modeling approaches have certain advantages, as briefly discussed in the introduction. First, generated models can be directly executed. For instance, Navarre et al. [18] present their Interactive Cooperative Objects (ICO) approach, which is based on Petri nets. Using their interpreter called PetShop [19], generated models can be directly executed or simulated. ICO models mainly describe interaction logic in the sense discussed above. Barboni et al. [20] extended the ICO approach with a graphical user interface markup language, called UsiXML to define also the physical representation in a user interface model. UsiXML [21] is an XML-based user interface description language, that offers a "multi-path development" process, enabling the user to describe a user interface on different levels of abstraction. Still, UsiXML primarily defines the physical representation and only specifies, which sort of functionality is connected to it without describing it. Among others, the User Interface Markup Language (UIML) [22] is another XML-based markup language for describing user interfaces, which also excludes interaction logic from its description. Further formal modeling approaches can be found, such as the Petri net-based approach described by de Rosis et al. [23] or by Janssen et al. [24].

The second argument for the use of formal models is verification, using for instance, model checking or other formal verification methods. Brat et al. [25] discuss an approach using model checking to verify and validate formal descriptions of dialogs. This is of main interest, e.g., in modeling of user interfaces in safety critical situations [26]. Furthermore, Paterno and Santoro [27] discuss the use of formal verification in context of the investigation of multi-user interaction.

Finally, formal models of user interfaces can be used to apply reconfigurations to it and thus change their outward appearance, behavior, or both without necessarily leaving the formalization. Navarre et al. describe in [28] and [29] the reconfiguration of formal user interface models based on predefined replacements that are used in certain situations in safety-critical application scenarios, such as airplane cockpits. Blumendorf et al. [30] introduce an approach that changes a user interface during runtime. This approach is based on so-called "executable models", which combine design information and the current runtime state of the system. Interconnections between system and user interface are changed appropriately during runtime. Another approach that applies reconfiguration during runtime has been introduced by Criado et al. [31].

In conclusion, adaptive user interfaces play a central role in human-computer interaction and are still an ongoing research activity. Formal techniques in the development, creation, and reconfiguration are still discussed in research literature, offering various advantages regarding modeling, execution, and verification. Petri net-based as well as XML-based approaches are already applied in various application scenarios. Nevertheless, none of these approaches presents a full-fledged approach for the creation and reconfiguration of user interface models in one coherent formal modeling approach. Furthermore, none of the presented approaches discusses a closely related concept that enables computer-based systems to generate and apply reconfiguration in a flexible and usecase independent way. Therefore, this work introduces a self-contained approach for visual modeling and creation, rule-based reconfiguration, and algorithmic rule generation of user interfaces that builds a formal framework for the creation of adaptive user interfaces.

III. Formal Modeling of User Interfaces

As has been argued above, formal modeling of user interfaces offers various advantages, such as closing the gap between modeling and implementation. Nevertheless, formalization is often related to the use of complex description languages and requires a deep understanding of the whole formalization concept. The latter is addressed by solid documentation, which still needs a basic expert knowledge of a certain domain. In case of user interface modeling as introduced here, the modeler should have a basic understanding of programming languages and process modeling. Still, the problem of learning how to use a modeling language is mitigated by the use of a visual language paired with an intuitive point-and-click editor implementation.

The gap between modeling and execution is finally closed by a transformation of a given user interface model into reference nets, a special type of Petri nets. Using its associated simulator implementation (called Renew [32]), the whole model becomes executable, while rendering of the physical representation of the user interface is supported by a further software component, as implemented in the UIEditor (further discussed in Section III-E).

Another advantage of a formal representation of a user interface model is the possible close integration of reconfiguration concepts. This is achieved by using a rewriting concept applicable to reference nets. According to various reasons (as further discussed below in Section III-D), graph rewriting based on category theory has been chosen. Using this kind of rewriting, the rewritten user interface model does not have to be transformed in anyway reducing possible loss of information in the transformation. Finally, the rewriting

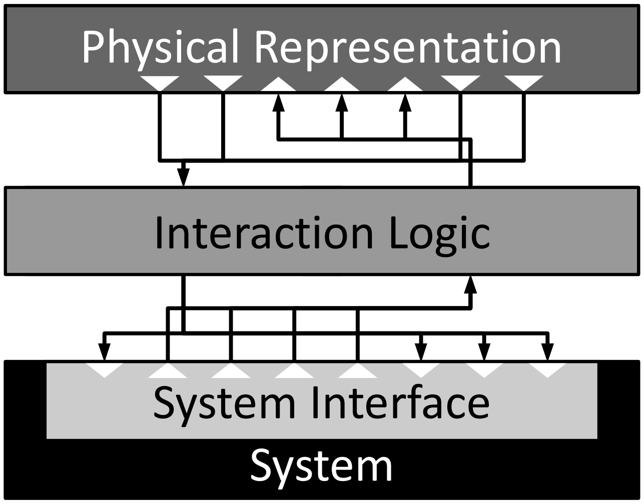


Figure 1. A two layered user interface model: the physical representation is directly accessed by the user, where the interaction logic specifies the data processing between physical representation and system to be controlled.

process, as well as the rewritten model is still verifiable in certain boundaries.

Therefore, this section introduces a formal modeling approach for user interfaces on a simple architectural basis, which is associated to a transformation algorithm that generates a reference net out of a created user interface model. After discussing a small example of the introduced modeling language and the transformation algorithm to reference nets, its associated reconfiguration concept will be introduced in more detail. Finally, this section will describe a tool called UIEditor that implements the modeling and reconfiguration concepts for user-driven, interactive creation and reconfiguration of user interface models.

A. Formal Interaction Logic Language - Formal Syntax

The basic concept of our approach for formal modeling of user interfaces relies on a two layered architecture that differentiates a user interface into its *physical representation* and its *interaction logic* (cp. Figure 1). In general, the term physical representation is not restricted to classic graphical user interfaces (GUIs) or WIMP (Windows, Icons, Menus, Pointers) interfaces [33]. Thus, a physical representation could also be a combination of speech recognition as input and a haptic device for output (this combination relates to multimodal user interfaces, such as described in [34]). Nevertheless, our work focuses on graphical user interfaces involving classic interaction elements, such as buttons, sliders, or text fields, as a first step implementation of the approach.

Interaction logic specifies the logical behavior of a user interface. It is defined by a set of processes that specify how data is processed that is emitted from the physical representation, such as events or inputted text, or from the system to be controlled. The system to be controlled can be specified as third layer but is not part of the user interface model (such as can be seen in Figure 1). Thus, interaction logic specifies the data processing, which takes place between the physical representation and the system to be controlled. These processes can be understood as graphs specifying data flow and data processing, also called *interaction processes*. Certain nodes in these graphs are dedicated to connect the process to the system or to the physical representation. Other nodes encapsulate complex data processing operations, such as casting of data types or arithmetic operations, and so forth.

We developed a visual and graph-based formal modeling language called *Formal Interaction Logic Language* (FILL) to support easy modeling capabilities for creating and editing interaction logic models in a visual editor (see Section III-E). Thus, FILL fulfills the requirement of providing visual modeling capabilities for creating interaction logic. First, the formal definition of FILL's syntax is given below (cp. [7]), which is partly based on nodes defined in the Business Process Model and Notation language (BPMN) [35].

Definition 1: the *Formal Interaction Logic Language* (FILL) is a 19-tuple

$$(S, I, C_I, C_O, P_I, P_O, X_I, X_O, B, T, P, E, l, g, c, t, \omega, \mathcal{L}, \mathcal{B}),$$

where S is a finite set of system operations, and I is a finite set of interaction-logic operations; P_I and P_O are finite sets of input and output ports; X_I and X_O are finite sets of input and output proxies; C_I is a finite set of input channel-operations; C_O is a finite set of output channel-operations; B is a subset of BPMN-Nodes, with

$$B = \{\oplus,\ \otimes,\ \odot\}. \tag{1}$$

S, I, C_I, C_O, P_I, P_O, X_I, X_O, T, and B are pairwise disjoint.

P is a finite set of pairs

$$\begin{aligned} P = \{(p,o) \mid p_I(p) = o\} \cup \{(p,o) \mid p_O(p) = o\} \cup \\ \{(p,o) | p'_I(p) = o\} \cup \{(p,o) | p'_O(p) = o\}, \end{aligned} \tag{2}$$

where $p_I : P_I \rightarrow S \cup I$ and $p_O : P_O \rightarrow S \cup I$ are functions with

$$\begin{aligned} \forall s \in S : (\exists_1 (p,s) \in P : p_I(p) = s) \wedge \\ (\exists_1 (p,s) \in P : p_O(p) = s),\ and \end{aligned} \tag{3}$$

$$\forall i \in I : \exists_1 (p,i) \in P : p_O(p) = i, \tag{4}$$

and where $p'_I : P_I \rightarrow C_I$ and $p'_O : P_O \rightarrow C_O$ are functions with

$$\begin{aligned} \forall c \in C_I : (\exists_1 (p,c) \in P' : p'_I(p) = c) \wedge \\ (\nexists (p,c) \in P' : p'_O(p) = c),\ and \end{aligned} \tag{5}$$

$$\begin{aligned} \forall c \in C_O : (\exists_1 (p,c) \in P' : p'_O(p) = c) \wedge \\ (\nexists (p,c) \in P' : p'_I(p) = c). \end{aligned} \tag{6}$$

E is a finite set of pairs, with

$$\begin{aligned} E = \{(p_O, p_I) \mid e(p_O) = p_I\} \cup \\ \{(p,b) \mid e'(p) = b,\ b \in B\} \cup \\ \{(b,p) \mid e'(b) = p,\ b \in B\}, \end{aligned} \tag{7}$$

where $e : P_O \cup X_O \rightarrow P_I \cup X_I \cup \{\omega\}$ is an injective function, ω is a terminator, and

$$\begin{aligned} \forall (p_O, p_I) \in E : (p_O \in X_O \Rightarrow p_I \notin X_I) \wedge \\ (p_I \in X_I \Rightarrow p_O \notin X_O), \end{aligned} \tag{8}$$

and where

$$e' : P_O \cup X_O \cup B \to P_I \cup X_I \cup B \cup \{\omega\} \tag{9}$$

is a function, extending e from basic FILL, and

$$\begin{aligned}\forall b \in B : (\#\{(p,b)|(p,b) \in E'\} > 1 \Rightarrow \exists_1 (b,p) \in E') \\ \vee(\#\{(b,p)|(b,p) \in E'\} > 1 \Rightarrow \exists_1 (p,b) \in E').\end{aligned} \tag{10}$$

l is a function with

$$l : E' \to \mathcal{L}, \tag{11}$$

where $\mathcal{L}$ is a set of labels.
g is a function with

$$g : B \to \mathcal{B}, \tag{12}$$

where $\mathcal{B}$ is a set of Boolean expressions, also called guard conditions or guard expressions.
c is a relation with

$$c : C_I \to C_O. \tag{13}$$

T is a finite set of data types and t is a total function with

$$t : (P_I \cup P_O \cup X_I \cup X_O) \to T. \tag{14}$$

The visual representation of FILL's elements (syntax) is shown in Figure 2. FILL is mainly divided into four kinds of nodes (operation nodes, proxy nodes, BPMN nodes, and a terminator node) and two types of edges (data flow edge and channel reference edge), which are named according to the previous given syntax definition. *Operation nodes* are nodes that specify connections to the system (*system operation*), represent data processing operations (*interaction-logic operation*), or define relations between different subgraphs of the interaction logic (*channel operation*). Operation nodes are in general equipped with input and/or output ports. These connection points for edges represent data input or output. For instance, the shown example of an interaction-logic operation in Figure 2 consists of three input ports and one output port. Thus, the semantic of this operation is that it is executed if three data objects are sent to the three input ports via incoming edges. After the data processing function or method associated to the operation has been executed successfully, the result is passed back into the process via the single output port. In case of system operations, the inputted data object is passed to the system and/or a data object is returned from the system into the interaction process. Channel operations are further connected to each other by channel reference edges. Data objects sent to an input channel operation are redirected to one or more output channel operations as defined by channel reference edges. This enables FILL models to be modularized.

Another group of nodes are *proxy nodes* (Figure 2, upper right corner). These represent interaction elements that are part of the physical representation. Thus, proxy nodes are capable of sending data objects to the interaction process emitted by an interaction element or returning a data object from an interaction process to the associated interaction element. *BPMN nodes* as third group (Figure 2, right) define fusion and branching of interaction processes. Every node follows another type of fusion and branching semantics. An *AND* node branches an incoming interaction process by sending a copy of the incoming data object to every outgoing interaction process. In case of fusing different interaction processes, the outgoing process will be only triggered if all incoming interaction processes provide a minimum of one data object. An additional guard condition has to define which incoming data object will be copied to the outgoing interaction process, as can be seen in Figure 2. An *XOR* node has a contrary semantic to an AND node. In the branching case, exactly one outgoing process will be triggered by an incoming data object, which has to be further specified by a guard condition. In the fusion case, the XOR node simply redirects an incoming data object (whatever data process sent this data object) to the outgoing process. The *OR* node is a mixture of both AND and OR nodes. By specifying groups of incoming (fusion case) or outgoing (branching case) edges, an OR node behaves as an AND node concerning groups (every edge of a group has to provide a minimum of one data object in the fusion case, or every process of a group is triggered in the branching case, respectively), where edge groups are handled similarly to an XOR node among each other.

As far as formal languages are defined by formal syntax and semantics, formal semantics can be defined in two ways: (a) define the semantics of a formal language using mathematical formalism or (b) formally map a language's syntactic elements to another formal language that provides formal semantics. In case of FILL, a formal transformation to reference nets has been defined and algorithmically implemented. A reference net is a special type of Petri net, which has been equipped with formal syntax and semantics definitions [36]. It is a colored Petri net formalism that specifies an inscription language offering the definition of typed tokens and the specification of references to net instances. This mechanism makes it possible to instantiate nets and to assign resulting net instances to tokens using references. Furthermore, transitions can be inscribed with synchronous channels, which can be also used to call Java methods using the associated simulator Renew [32], which is implemented in Java. Java is further used to transform FILL to reference nets, where interaction-logic and system operations are implemented based on pre-defined interfaces. This enables Renew to call these functions through the interfaces' implementations. In general, a synchronous channel associates transitions with each other in such a way that they are only able to fire synchronously, which is also true for associated Java methods as discussed above.

B. Formal Interaction Logic Language - Formal Semantics through transformation to reference nets

The transformation of FILL to reference nets is algorithmically defined. The whole formal specification of the algorithm can be found in [7]. In this paper, the transformation will be discussed visually because the formal definition of the algorithm would exceed the scope of the paper.

Before starting the description of the transformations, some definitions are necessary to understand the inscriptions generated by the algorithm. To stay consistent to the original specification of the transformation algorithms, the definitions below have been extracted from the original sources [37] and [7] respectively.

Definition 2: Assume a given FILL graph as 19-tuple as defined above. Based on this, the functions f, κ, id, id_s can be defined as follows.

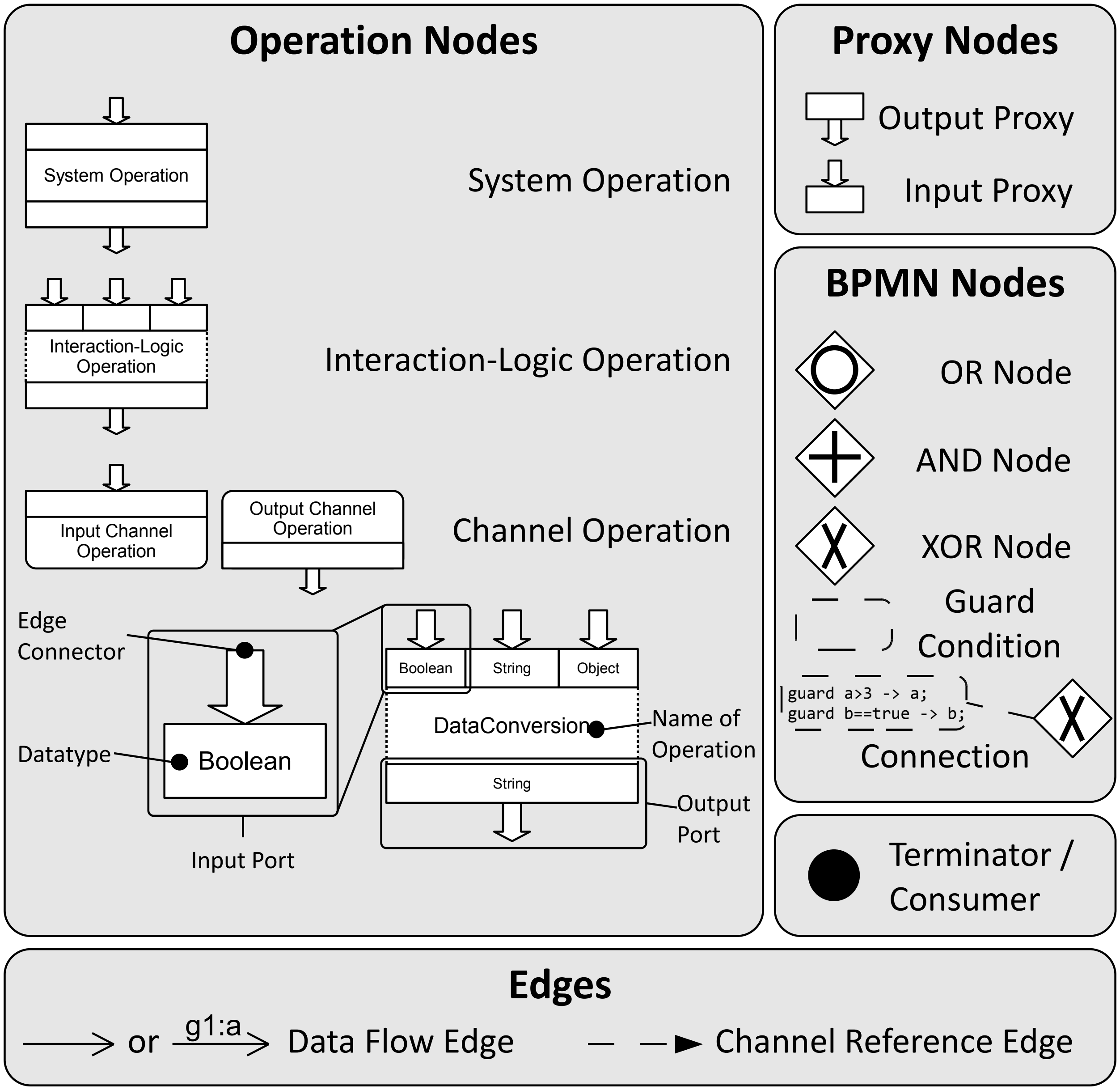


Figure 2. Visual specification of syntactical elements of the Formal Interaction Logic Language (FILL).

f is a function, with

$$f : S \cup I \cup C_I \cup C_O \to \mathcal{F}, \tag{15}$$

where $\mathcal{F}$ is a set of function calls. These function calls reference different types of underlying structures in the system or in the implementation of interaction-logic operations. Depending on the underlying programming language or system, these references have different syntaxes. Here, reference nets use Java method calls for calling code from the net.

κ is a function, with

$$\kappa : X_I \cup X_O \to \mathcal{I}, \tag{16}$$

where $\mathcal{I}$ is a set of references on interaction elements on the physical layer of the user interface.

id is a total bijective function, with

$$id : S \cup I \cup C_I \cup C_O \cup P_I \cup P_O \cup X_I \cup X_O \cup B \to \mathcal{ID}, \tag{17}$$

where $\mathcal{ID}$ is a set of ids, that identifies any node, port, or proxy in FILL. Based on the formal, graph-based definition of FILL, global identifiers are not necessary. In the transformation to reference nets and for representation in data formats like XML, ids play an important role.

$id_s : S' \to \mathcal{ID}$ is a total bijective function that matches a place in a reference net to an id similar to function id. $S' \subseteq S$ is a subset of places representing connections to and from a BPMN node. This function is necessary for the transformation of BPMN nodes; it compensates for the fact that a BPMN node does not have ports associated with ids. The inverse function id_s^{-1} matches an id to a place in a reference net. Due to the

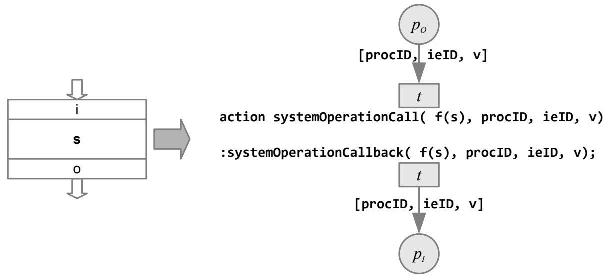


Figure 3. Transformation of a system operation into a reference net.

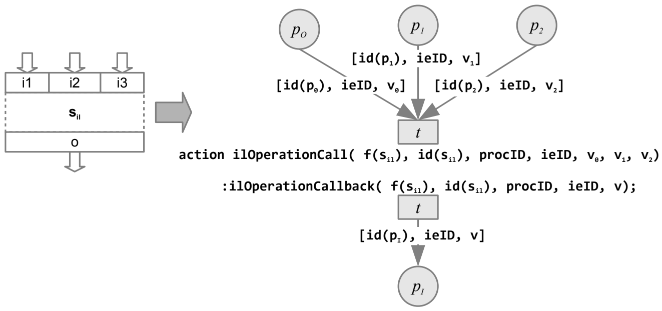


Figure 4. Transformation of a interaction-logic operation into a reference net.

bijectivity of id_s, there is an inverse function id_s^{-1}.

$ieID$ and $procID$ are ids that are generated in the transformation process. $ieID$ indicates the id associated to the interaction element that triggers or is triggered by the interaction process. $procID$ is used to further specify the data flow, as can be seen in the example discussed in Section III-C.

1) Transformation of Operation nodes: The transformation of interaction-logic or system operation nodes basically results in the generation of two transitions; one calling an associated (Java) method and one for reentering the net after the method returns. The inscriptions of these transitions only differ in the naming of the synchronous channel, which calls the associated method (systemOperationCall vs. ilOperationCall) and specifies the reentering point (systemOperationCallback vs. ilOperationCallback). They further differ in the number of variables that are send to the method (see Figure 3 and 4, respectively), which is indicated by its name $f(op)$, where op indicates the transformed operation node. Data values sent to and from operation nodes are associated to variables, here indicated with v and v_0 to v_2 in Figure 3 and 4.

The main difference between the transformation of a system operation node (as shown in Figure 3) and the transformation of an interaction-logic operation node (as shown in Figure 4) is the transformation of input ports. According to FILL's syntax definition, every interaction-logic operation has 0 or 1 output port and 0 to n input ports. In case of system operation nodes, there is exactly 1 input and 1 output port. In general, input and output ports are transformed into an edge/place combination as can be seen in Figures 3 and 4.

For the transformation of channel operations, channel edges have to be considered beside the operation nodes themselves. First of all, output channel operations are transformed into a transition-edge-place subnet as can be seen in the lower left corner of Figure 5. The transformation of an input channel operation is more complex. For every outgoing channel edge (connection), a place-edge-transition subnet (indicated as places q_0 to q_2 and transitions t_0 to t_2 in Figure 5) is generated, which is further connected to a main transition representing the operation (indicated as t_I in Figure 5). The connection between input and output channel operations is transformed into inscriptions, such as shown in Figure 5 for c_I and c_0, which are indicated by the used id of c_0 retrieved by $id(c_0)$. Both transitions are connected via a synchronous channel named $channel$. The keyword $this$ references to the current net instance, thus does not reference another net instance or external sources. If the shown example net is simulated, transition t is fired synchronously with transition t_0 according to the synchronous binding semantics of synchronous channels in reference nets.

2) Transformation of BPMN nodes: For BPMN nodes, the transformation into reference nets focuses even more on the structure of the generated net than it is the case for operation nodes. Here, the firing semantics of reference nets is actively used for modeling of the semantics of BPMN nodes as used in FILL. The semantics of BPMN nodes in FILL has been discussed above in Section III-A. Below, the transformations will be described per BPMN node in case of fusion and branching of interaction processes. For any transformation it is true that for any incoming and outgoing process a place is generated defining the entrance or exit point of the BPMN node, as can be seen in Figure 7.

AND(fusion): The outgoing process is only triggered if all incoming processes provide one or more data objects. This semantic is reflected in the structure of the reference net by defining the places representing the incoming processes as precondition for the transition t, which represent the BPMN node itself. The associated guard condition specifies which data object (here the object associated to the variable a) is copied to the outgoing process. In this case, the guard condition is obligatory in the FILL graph. The syntax of guard conditions has been specified compatible to guard conditions in reference nets, as specified in [36].

AND(branch): All outgoing processes will be triggered if the incoming process provides a data object. This semantic has been simply realized by specifying all places representing an outgoing process as postcondition of the transition representing the AND node. The shown guard condition in Figure 7 is optional and specifies under which condition the incoming data object is redirected to the outgoing processes.

XOR(fusion): Every incoming data object will be redirected to the outgoing edge by copying the data object to it. Therefore, for every incoming process one transition is generated that is optionally inscribed by a guard condition corresponding to the guard condition specified in the genuine FILL graph.

XOR(branch): Only one of the outgoing processes is triggered in case of an incoming data object. Therefore, any outgoing process is represented as a transition in the transformed reference net. Which outgoing process will be triggered has to be defined by an obligatory guard condition in the same sense as discussed above.

OR(fusion): Groups of processes can be defined by edge inscription in the FILL graph as can be seen in Figure 7.

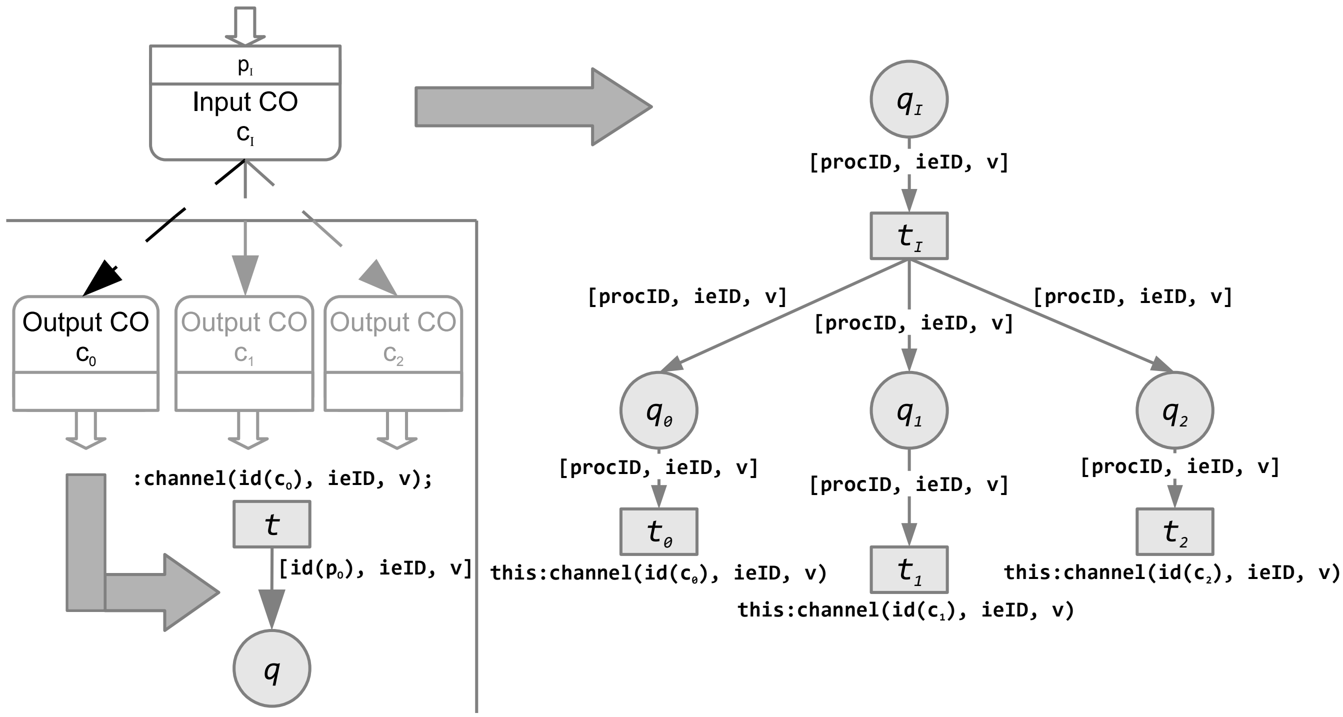


Figure 5. Transformation of channel operations into a reference net.

Subsequently, the transformation generates an AND like subnet for every group and thereby behaves like an XOR node between the groups. If all incoming processes of one group provide a data object each, the group's associated transition can fire, independent from other groups. Guard conditions control which data objects are redirected to the outgoing process.

OR(branch): Outgoing process groups are triggered according to the guard condition defined in the FILL graph. The assignment of the guard conditions to the group is defined by an arrow in the FILL graph's guard condition, as it is the case for all guard condition assignments for edges in the above cases of *AND* and *XOR* nodes.

3) Transformation of proxy nodes: Proxy nodes represent data connectors to and from interaction elements. Therefore, κ is a function relating a proxy node to its associated interaction element by a unique reference. This reference is used as specification of a channel name in case of an output proxy node (see Figure 6 left), such that an event can be uniquely redirected to the correct proxy node representation in the reference net. The callback function from the net to the physical representation and the associated interaction element is specified by a fixed channel name called `widgetCallback`. To identify the correct interaction element on the side of the physical representation, its identifier (given by κ) is passed as parameter to the channel (see Figure 6 right).

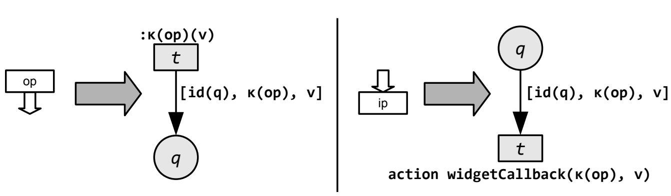


Figure 6. Transformation of proxy nodes into a reference net.

The section below will present a comprehensive example of the use of FILL and an associated transformation to reference nets. The example provides a deeper insight to the semantics of a FILL graph and how an associate reference net transformation looks like before Section III-D introduces the rewriting approach for interaction logic models.

C. FILL Example

Figure 9 shows a FILL graph that consists of two interaction processes; on the left an interaction process is shown using an XOR BPMN node, on the right a simple interaction process that triggers a system operation is shown. The latter process starts with an interaction-logic operation that only supports a single output port without any input port. Thus, the operation only emits data objects into the process but does not consume any. In this case, the operation called "ticker" sends a simple object into the interaction process and thereby triggers the following system operation called "getSV2Status". This system operation returns the associated value, here the status of SV2 that represents whether a steam valve of a simple nuclear power plant simulation is open (true) or closed (false). This value is then sent to an input proxy connected to an interaction element, such as a lamp widget that flashes green in case of a true value and red in case of a false value.

Before discussing the interaction processes in more detail, the simple nuclear power plant simulation will be briefly presented. In Figure 8, the process is shown. The nuclear power plant simulation consists of three main elements: the reactor, the condenser, and the turbine, which transfers steam into rotation energy that is further transferred into electrical energy

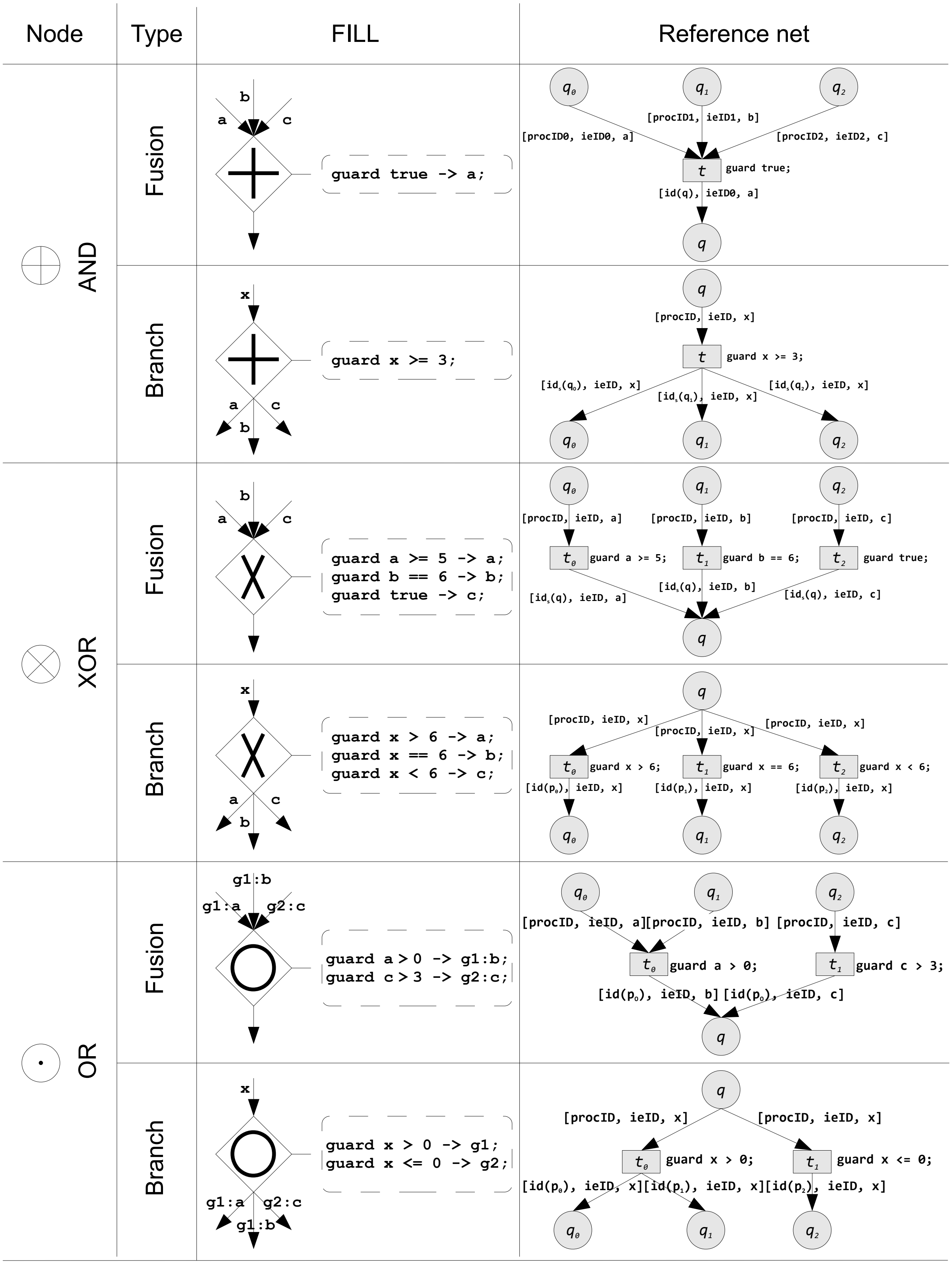


Figure 7. Transformation of BPMN nodes into a reference net.

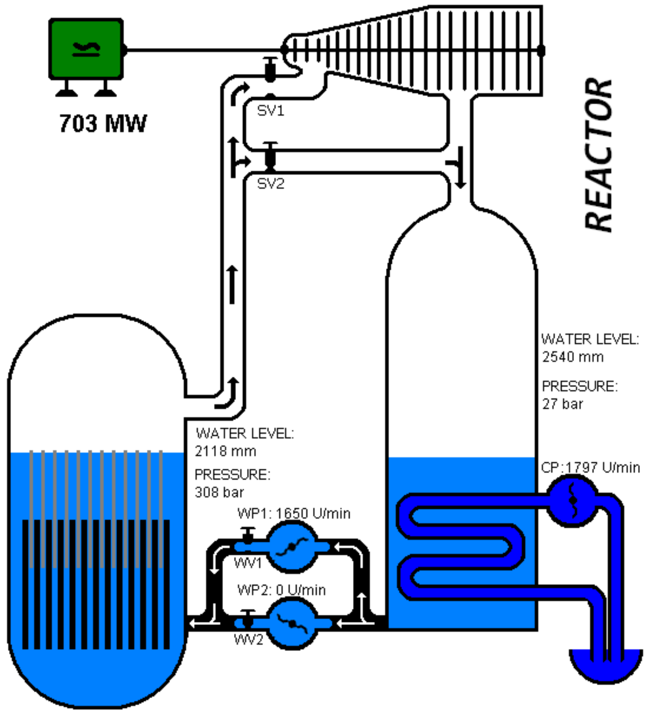


Figure 8. Feed water circuit of a simplified simulation of a steam water nuclear reactor called *REACTOR*.

through a connected generator. The condenser condenses the steam back into fluid water that is pumped by two pumps back into the reactor. The nuclear reaction generates thermal energy and is controlled by control rods, which can be pushed into or removed from the nuclear core. Removing increases the amount of thermal energy boiling the water and thereby generating steam. Various valves can be further used to control the way of the water and the steam.

The process on the left of Figure 9 also includes an XOR BPMN node branching the interaction process into two sub-processes. The whole process is triggered by an interaction element, like a button, sending an event object to the interaction process. Before the XOR node is triggered, the same system operation is triggered as in the left interaction process, such that the status of the valve is sent to the XOR node. The associated guard condition specifies that in case of a false value, the sub-process indicated with `a` will be triggered by the inputted value. In case of a true value, the sub-process `b` will be triggered. In both cases, an interaction logic that generates a Boolean value and emit this to the system operation “setSV2Status” is executed. Thus, if the current value of the steam valve is true, it will be changed to false and vice versa.

The result of the transformation to a reference net is shown in the middle of Figure 9. Gray boxes with different types of borders indicate which FILL element is transformed into which subgraph of the reference net. Furthermore, it can be seen how edges in the FILL graph are transformed into transitions, simply redirecting incoming data objects to their outgoing edges. `procIDs` are used to specify a certain interaction process throughout the reference net. This is of special interest in case of transformation of system operations, which have only one representation in the reference net. This has various reasons. System operations can influence or return (part of) the system state; these operations are state-full. Write-write race condition should be avoided in case of state-full software components and thereby only single representations of system operations exists. Please note that the presented example intends to give a deeper insight how a result looks like that the algorithm generates.

Therefore, procIDs are necessary to identify the correct reentering point after returning from a system operation. In Figure 9, this can be seen in case of the system operation “getSV2Status”. Here, the relevant procIDs are marked with dashed ellipses.

Interaction-logic operations are in contrast to system operations transferred into multiple sub-nets in the reference net. For every used interaction-logic operation block in the FILL graph, a subgraph is created. In this case, the above discussed concurrent method calls could occur. Still, interaction-logic operations should not be state-full. Thus, the race conditions as described above will not occur. Additionally, pre-defined parameters, such as the parameter specifying which Boolean value should be generated is different between every use of the interaction-logic operation. In Figure 9, the operation “generateBooleanValue” is called once with the parameter `new Boolean(true)` and once with the parameter `new Boolean(false)`.

D. Formal Reconfiguration

Adaptive user interfaces offer great opportunities in human-computer interaction (what has been discussed in detail, above). Thus, formal user interface models should be enabled to be adaptive or even adaptable in a certain sense. This section introduces a formal reconfiguration concept that is able to adapt the presented graph-based user interface modeling approach.

Formal reconfiguration can be differentiated from redesign, where redesign refers to the change of the physical representation and reconfiguration specifies changes in the interaction logic of a user interface model. Interaction logic is modeled using FILL and then transformed to reference nets, not only for defining formal semantics but also for making FILL graphs executable. Thus, reconfiguration means changing reference nets, necessitating a method (a) that is able to change reference net models and (b) that is defined formally to prevent reconfigurations from being non-deterministic. Various graph transformations and rewriting approaches can be found in literature. Shürr and Westfechtel [38] identify three different types of graph rewriting systems. The logic-oriented approach uses predicate logic expression to define rules. This approach is not wide-spread due to its complex implementation. Another approach defines rules based on mathematical set theory, which is flexible and easily applied to various applications. Still, it has been shown that irregularities could occur applying set-theoretical rules to graph-based structures.

Finally, graph rewriting based on category theory has been chosen for reconfiguration according to various features. First of all, pushouts (see Definition 3) as part of category theory are well behaved regarding their application to graphs, especially the double-pushout (DPO) approach as has been discussed by Ehrig et al. [39]. The DPO approach specifies rules that

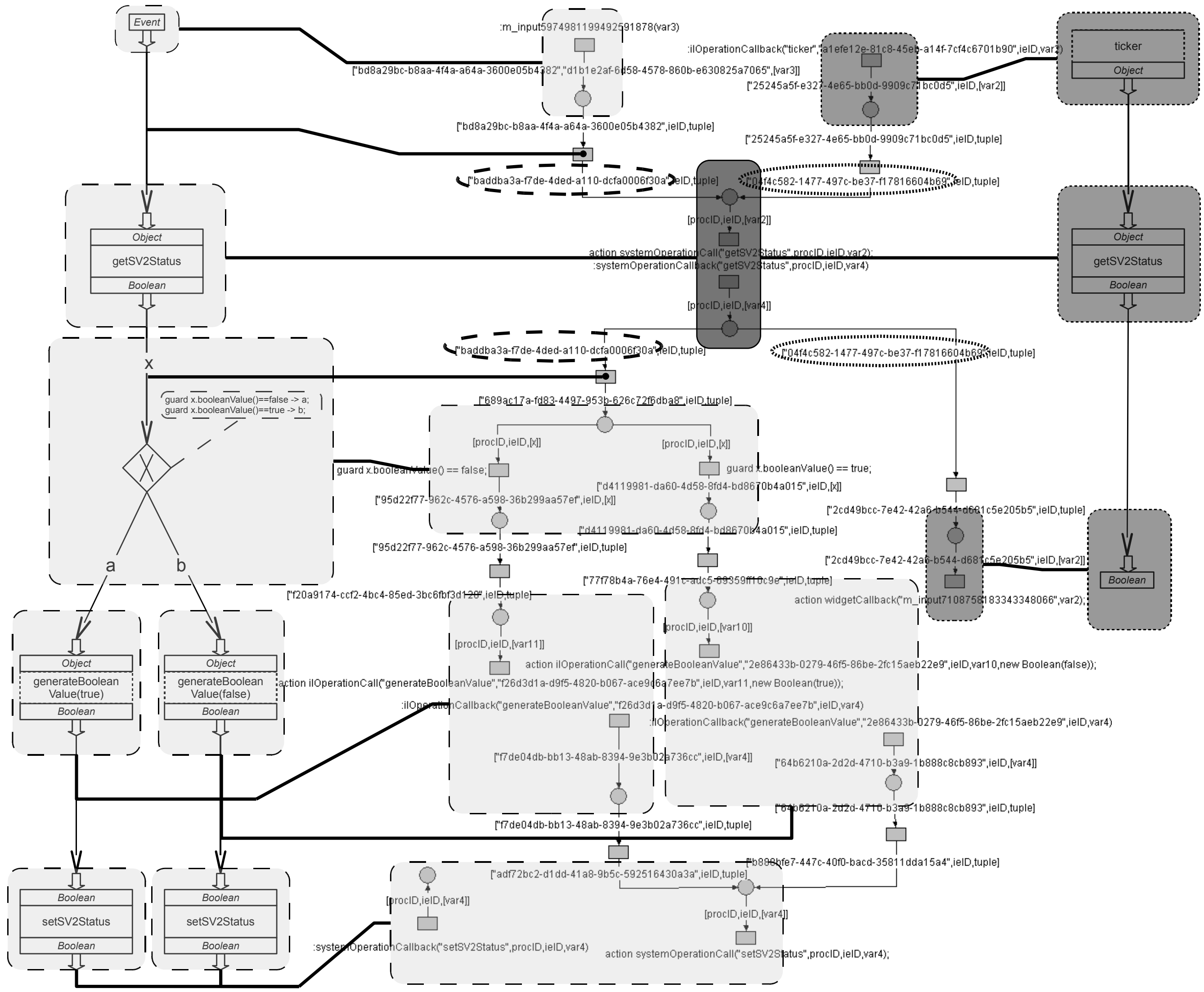


Figure 9. An example of a FILL graph with two interaction processes, showing their corresponding transformation to a reference net.

explicitly define which nodes and edges are deleted in a first step and then being added to the graph in a second. This is not true for the single-pushout (SPO) approach, which is implementation dependent or generates results that are probably not valid graphs [39]. Only to give a simple example, the SPO approach can afford dangling edges, which are edges having only a source or a destination, but not both. A further problem could be the implicit fusion of nodes, which could have negative implications to the rewriting of interaction logic. These aspects have been resolved in the DPO by making deletion and adding of nodes and edges explicit as well as defining a condition preventing rules from being valid if they produce dangling edges. The DPO approach definition will make this more precise, as given below.

A further argument supporting the use of the DPO approach for rewriting interaction logic is that it has been extended and discussed in context of Petri nets as discussed by Ehrig et al. in [40] and [41]. This work offers a solid basis for the reconfiguration of reference net-based interaction logic. Furthermore, the DPO approach (as well as the SPO) is able to change existing graphs, where graph grammars are production systems. Using graph grammars for reconfiguration would mean to change production rules instead of defining rules changing an existing graph. At a first glance, this seems to be less comfortable and counter intuitive. Finally, the Petri net-based DPO approach as described by Ehrig et al. [40] can be simply extended to colored Petri nets. Within certain boundaries, also the semantics of the inscription can be taken into account, as described in detail by Stückrath and Weyers [42].

As the SPO, the DPO is based on the category theory-based concept called pushouts. Assuming a fundamental understanding of category theory (otherwise consider, e.g., [43]), a pushout is defined as follows.

Definition 3: Given two arrows $f : A \to B$ and $g : A \to C$, the triple $(D, g^* : B \to D, f^* : C \to D)$ is called a *pushout*, D is called *pushout object* of (f, g), and it is true that

1) $g^* \circ f = f^* \circ g$, and

2) for all other objects E with the arrows $f' : C \to E$ and $g' : B \to E$ that fulfill the former constraint, there has to be an arrow $h : D \to E$ with $h \circ g^* = g'$ and $h \circ f^* = f'$.

The first condition specifies that it does not matter how A is mapped to D, that is via B or C. The second condition guarantees that D is unique, except isomorphism. Thus, defining (f, g) there is exactly one pushout (f^*, g^*, D) where D is the rewritten result, also called *pushout object*. In general, A and B are given defining the changes applied to C, the graph to be rewritten. Therefore, a rewriting rule can be specified as a tuple $r = (g, f, A, B)$, such that D is the rewritten result by calculating the pushout (object). This procedure is mainly applied in the SPO approach.

For the definition of the DPO approach, the pushout complement has to be defined first.

Definition 4: Given two arrows $f : A \to B$ and $g^* : B \to D$, the triple $(C, g : A \to C, f^* : C \to D)$ is called the *pushout complement* of (f, g^*) if (D, g^*, f^*) is a pushout of (f, g).

A DPO rule is then defined based on the definition of a *production* corresponding to the former discussion of pushouts in category theory.

Definition 5: A *matching* is a mapping $m : L \to G$; a *production* is a mapping $p : L \to R$, where L, R, and G are graphs. The corresponding mappings of m and p are defined as mapping $m^* : R \to H$ and $p^* : G \to H$, where H is also a graph.

Definition 6: A *DPO rule* s is a tuple $s = (m, (l, r), L, I, R)$ for the transformation of a graph G, with $l : I \to L$ and $r : I \to R$, which are two total homomorphisms representing the production of s; $m : L \to G$ is a total homomorphism matching L to graph G. L is called the *left side* of s, R is called the *right side* of s, and I is called an *interface graph*.

Given a rule s, in a first step the pushout complement C can be calculated using L, I, m, and l with a given graph G to be rewritten. In the DPO approach, this step deletes nodes and edges from G. In the second step, the pushout is calculated using I, R, and r applied to C resulting in the graph H. This step adds nodes and edges to C. In conclusion, the difference between L and I specifies the part deleted from G, where the difference between I and R defines those elements, which are added to C and finally to G. The result of applying s to G is the graph H as can be seen in Figure 10.

Nevertheless, the pushout complement is not in all cases unique or probably does not even exists. According to the latter, if the total homomorphisms l and m fulfill the *gluing condition*, the pushout complement will always exist. The gluing condition is defined as follows.

Definition 7: There are three graphs $I = (V_I, E_I, s_I, t_I)$, $L = (V_L, E_L, s_L, t_L)$, and $G = (V_G, E_G, s_G, t_G)$. Two graph homomorphisms $l : I \to L$ and $m : L \to G$ fulfill the *gluing condition* if the following assertions are true for both l and m, given as

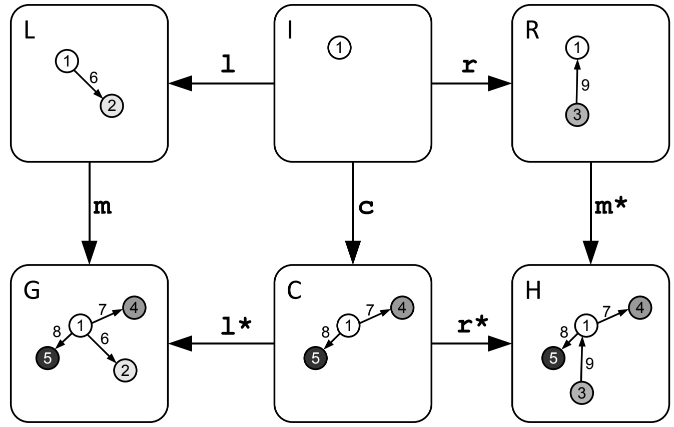


Figure 10. Exemplary DPO rule and its application to a graph G.

$$\nexists e \in (E_G \setminus m(E_L)) : s_G(e) \in m(V_L \setminus l(V_I)) \lor t_G(e) \in m(V_L \setminus l(V_I)), \tag{18}$$

and

$$\nexists x, y \in (V_L \cup E_L) : x \neq y \land m(x) = m(y) \land x \notin l(V_I \cup E_I). \tag{19}$$

Condition 18 is also called *dangling condition*. The homomorphism l of a DPO rule that defines, which nodes have to be deleted from a graph fulfills the *dangling condition* if it also defines which edges associated with the node will be removed. Thus, the dangling condition avoids *dangling edges*; a dangling edge is an edge that has only one node associated with it as its source or target. Condition 19 is called *identification condition*. The homomorphism m fulfills the identification condition if a node in G that should be deleted has no more than one preimage in L. However, if one node of G has more than one preimage in L defined by m and one of these has to be deleted, it is not defined whether the node will still exist in G or must be deleted. This confusion is avoided by the identification condition.

The problems of the SPO approach discussed above are mainly solved by the gluing condition being an integral part of the DPO approach. Nevertheless, the pushout complement is not unique but exists if the gluing condition is fulfilled. If l and m are injective, the pushout complement will be unique except isomorphism. This aspect is further discussed by Heumüller et al. [44] and in [7, p. 107].

The above definition of the DPO approach is only applied to simple graphs (cp. Figure 10). An extension of this approach to (simple PT) Petri nets has been discussed by Ehrig et al. in [40] and [45], and Weyers [7]. Nevertheless, rewriting of inscriptions has not been considered by these authors. Inscriptions extend basic Petri nets with further semantics, such as supporting complex data objects as tokens and the definition of guard conditions that extends the firing semantics of transitions. Still, rewriting interaction logic means rewriting reference nets, which are finally Petri nets with a specific inscription language that supports guard conditions and the definition of synchronous channels. Therefore, rewriting inscriptions has been discussed by Stückrath and Weyers in [42].

Finally, rule descriptions have to be serialized in a computer readable form. This is necessary for the implementation-side use of rewriting rules using the algorithmic rule generation concept presented in the next section. Therefore, an existing XML-based description language for Petri nets (PNML [46]) has been used with an extension to structure and describe the rule specific elements, such as indicating L as `deleteNet`, I as `interface`, and R as `insertNet`. The `mapping` node specifies m as a set of `mappingElements`, which are representations of tuples of XML ids. An example showing a rule in XML format can be seen in the XML snippet included in Appendix A.

E. The UIEditor

The above sections introduced a formal modeling and rewriting approach for the creation and reconfiguration of user interfaces, especially for the description of interaction logic. This approach is accompanied with a tool implementation called UIEditor, which offers capabilities for the creation/modeling, execution, and reconfiguration of user interface models. Figure 11 illustrates a general workflow implemented by the UIEditor that characterizes the creation and usage of user interface models based on our approach. In the *creation phase*, the user interface model is interactively created by an expert using the visual editor for editing the physical representation and the interaction logic using FILL. Figure 12 shows the UIEditor in creation mode with two opened visual editors. On the right, the drag-and-drop editor for the creation of the physical representation can be seen. On the left, the editor for creating the interaction logic using FILL is shown. Generally, FILL nodes and interaction elements are added to the canvases via drag-and-drop. In the case of FILL nodes, each node is connected via edges from output to input ports in an interactive fashion. Parameters of interaction elements can be specified through context menus and associated dialogs.

For the execution of user interface models, the UIEditor implements the transformation algorithm discussed in Section III-B, which is encapsulated in the creation of simulation files. These files define the connection of reference net-based interaction logic models, the description of the physical representation, the associated system to be controlled, and some further parameters like logging. After finishing the transformation, the UIEditor changes into simulation mode loading the simulation file and starting the execution of the user interface. In *simulation mode*, the UIEditor shows only the physical representation of the user interface model and starts the underlying controlled system. The interaction logic is loaded to Renew. The UIEditor handles the communication between reference net simulation, the physical representation, and the system to be controlled. Here, the user is able to interact with the system using the modeled and executed user interface.

If the user stops the simulation (by pressing the stop button), the UIEditor changes into *reconfiguration mode*. Here, the physical representation is shown to the user with a set of buttons related to the execution of reconfiguration rules. Various reconfiguration operations are supported, such as parallelization of input operations or discretization of interaction elements. In the first case, the user selects n different interaction elements and applies the parallelization operation to it. A new button is generated that executes all operations in parallel that were previously connected to the selected interaction elements. Discretization is a reconfiguration operation that generates a button, which sends a predefined value to an interaction process that has been previously connected to a continuous interaction element, such as a slider. In the current implementation, these operations generate DPO rules for changing the underlying interaction logic based on an algorithmic and hard-coded implementation. This makes the reconfiguration mechanism inflexible and hard to extend. Therefore, the next section will introduce a more flexible concept for describing and instantiating rules in a more flexible and extensible way using abstract description concepts and formats.

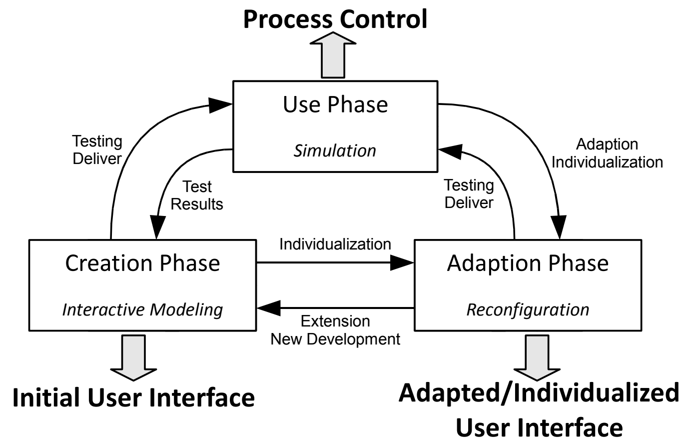


Figure 11. General workflow for using the UIEditor: creating user interface models, running these through simulation, and reconfigure these using an interactive tool.

IV. Rule Generation

The current implementation of reconfiguration in the UIEditor is exclusively user-driven and based on pre-defined algorithmic implementations for rule generation making the whole reconfiguration approach inflexible and hard to extend. Therefore, this section describes an extension based on the modeling and reconfiguration formalism introduced above (see Section III) according to (a) a generic and simple-to-use description of rule classes and (b) the algorithmic implementation of rule generation based on these classes. In general, the approach aims at the generation of DPO rules applied to reference net-based interaction logic as well as basic redesign rules applied to the physical representation, where the generation of DPO rules will be the focus of the discussion. The rule generation enables reconfiguration and redesign to be applied automatically as well as by the basic approach of user triggered reconfiguration, as discussed above. The whole approach is embedded into a rule generation process, which will be discussed in the following section. This process identifies necessary steps and algorithmic implementations for the intended rule generation, which will be presented in more detail in Section IV-B and IV-C. Finally, the approach will be discussed with an example of adaptive automation in Section V relating to the previous work discussed in [1].

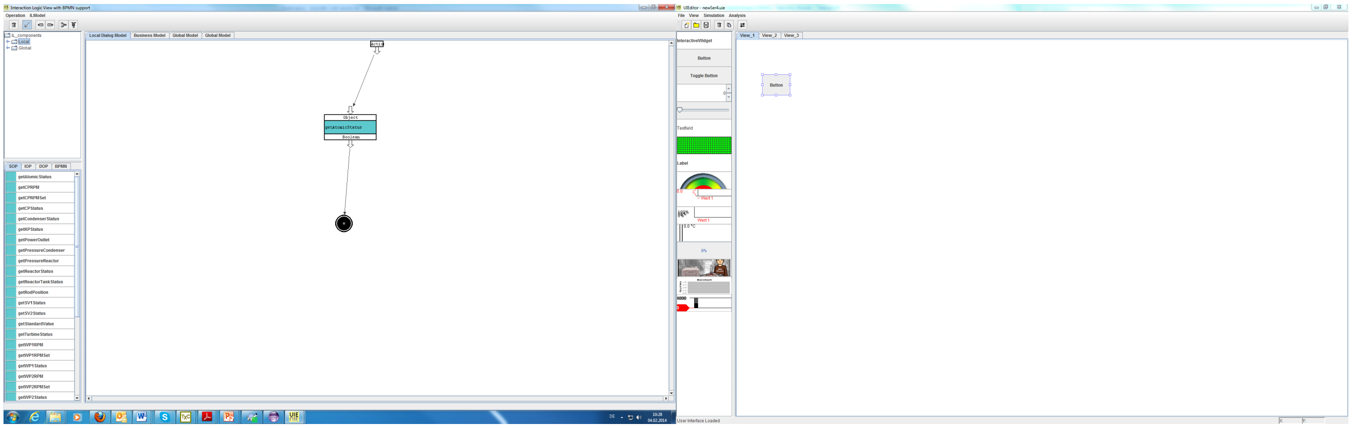


Figure 12. The UIEditor in creation mode: left, the interactive editor for modeling FILL graphs, right, the interactive drag-and-drop editor for modeling the physical representation.

A. Rule Generation Process

The rule generation process is illustrated in Figure 13. It specifies the main steps for selecting necessary information and data for the rule generation, certain validation and decision steps, and finally the rule generation and the rule application to the interaction logic and the physical representation of the user interface model.

First, the process has to be triggered, either by the user who interactively selects certain interaction elements and a rule to be applied or by the system that decides to adapt the user interface based on sensory data and certain domain knowledge. Still, this part of the reconfiguration process is not part of the approach discussed here regarding its domain and use case dependent implementation. However, this trigger operation ends up in a call of a user interface reconfiguration (indicated by ① in Figure 13) and in the gathering of input data, which is essential for further steps in the process. In step ①, a matching rule class is selected from a data base according to the information and data provided by the triggering instance. In step ②, the rule class gets validated based on various aspects. One major validation step here is the analysis of input data according to completeness and correct matching to the selected rule class, as well as the evaluation of the class' instantiation precondition. Therefore, rule classes specify preconditions according to data that has to be provided for a successful instantiation.

If the rule class is validated to be applicable, it is redirected to step ③ generating the rule. Otherwise, the process terminates. The rule generation step will be discussed in more detail in Section IV-B below. In general, the rule generation takes a rule skeleton as input and tries to create a set of rules using the genuine (interaction logic) net and the input parameters provided by the triggering instance. If the set is empty, no matching subnets in the genuine interaction logic could be found and the process terminates. If the set holds more than one rule, a rule has to be selected in step ④ using a certain heuristic. One simple approach would be to select one rule randomly or the first that is successfully generated. Alternatively, it could be also possible to select the rule that has the greatest impact on the rewritten net. Nevertheless, the used heuristic has to be specified in context of the rule class description, as will be further described in Section IV-C. After the rule has been applied to the genuine net in step ⑤ according to the DPO-based rewriting approach (discussed above in Section III-D), the rule class is checked according to necessary changes in the physical representation. If the physical representation has to be redesigned, the specified changes are applied to the user interface in step ⑥. Finally, the process terminates.

B. Rule Skeleton

Step ③ shown in Figure 13 generates rule instances from rule skeletons using input data and other parameters. This section will specify the structure of rule skeletons and the generation algorithm extracting rule instances from a given skeleton, which involves linking of resources into the skeleton. This is necessary to extract certain structures from the net to be rewritten in a second step. Rule skeletons are encapsulated into rule classes, which provide further information regarding validation, necessary data for instantiation, and directives for possible redesigns being applied to the physical representation of a user interface model. Rule classes will be discussed in more detail in Section IV-C.

A skeleton specifies three special types of graphs L, I, and R that represent a DPO rule, as well as the functions l and r (cf. Figure 10). Figure 14 shows a simple example of a rule skeleton. Here, a new interaction element (a slider)

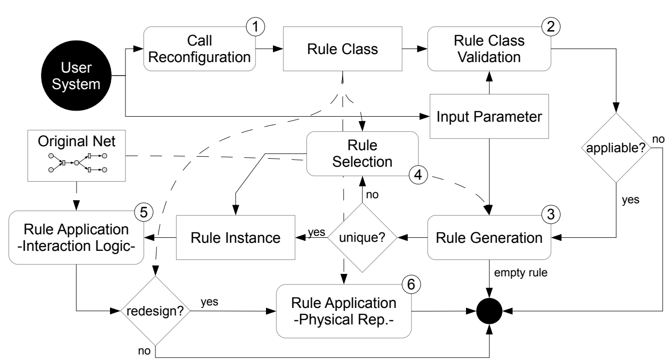


Figure 13. The rule generation process.

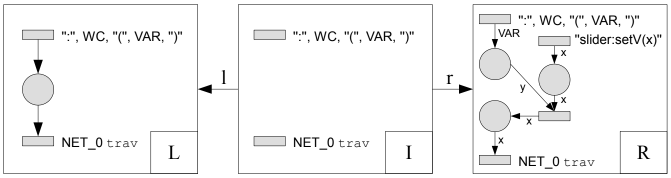


Figure 14. Simple example of a rule skeleton.

is added that is combined with an existing one, such that the existing element has to emit an event before the slider's value is redirected into the existing interaction process. The underlying system could, again, be the simple simulation of a nuclear reactor, as has been mentioned above (see Section III-C). In this case, the slider can be used to set the rounds per minutes or speed of a water pump in the system and then in a second step to set this value to the system by clicking the button.

1) Graph and text grammars: Rule skeletons are mixtures of graph and string grammars as well as control structures, such as loops or alternatives. Graph grammars are "... similar to a string grammar in the sense that the grammar consists of finite sets of labels for nodes and edges, an axiom, i.e., an initial graph, and a finite set of productions" [47, p. 120]. Graph grammars have been shortly presented above when discussing various rewriting systems (see Section III-D). In this context, graph grammars were not suitable because they are defined by rules that generate graphs and as opposed to change existing graph structures. In context of rule generation, grammatical descriptions are instead suitable because DPO rules need to be *generated* (and not rewritten) and thereby the graphs L, I, and R in particular.

In case of graph grammars, nodes inscribed with nonterminals are substituted during the generation by graphs or specific inscriptions, for instance extracted from the original graph through graph traversing or by given graph structures offered as initial parameter. Furthermore, inscriptions gets substituted by additional data or inscriptions extracted from the original data and the genuine net. Which data source is used for a substitution of nonterminals in the skeleton is specified in the rule class description or results from the expansion algorithm described below. The following list specifies which types of nonterminals can be used in rule skeletons, where EBNF refers to the Extended Backus-Naur Form, a special type of language for the definition of textual grammars:

- *EBNF like nonterminal symbols:* These nonterminals are used in inscriptions to be replaced by matching the associated node to a node in the genuine net (the net to be rewritten by the resulting rule) or by matching it to predefined parameters as has been inputted or specified by the rule class. In general, inscriptions are specified using EBNF syntax. Nonterminals are printed in capital letters only, such as `WC` in Figure 14.
- *Net nonterminal symbols:* Nodes inscribed by `Net` nonterminals getting replaced by (a) a subnet extracted from the genuine net or (b) by a predefined net given by the rule class or as additional input data. In case (a), the nonterminal in the skeleton is extended by a keyword specifying how the subnet has to be extracted from the genuine net. The example shown in Figure 14 (`NET_0 trav`) will be replaced by a traversed subnet of the genuine net. In case (b), the rule class has to specify by which net the nonterminal has to be replaced. In general, a net nonterminal is indicated by the keyword `NET_X`, where `X` specifies a further identifier of the nonterminal making it usable several times in the same skeleton.

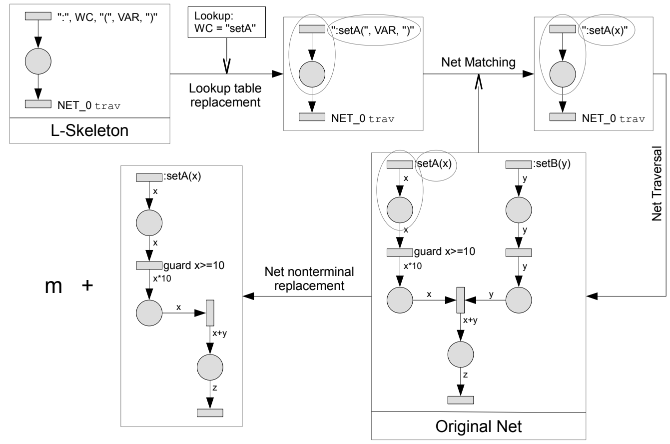


Figure 15. Example for nonterminal replacement by applying the REP algorithm.

For the instantiation of a rule skeleton, first nonterminals of the left side of the rule skeleton are replaced and the matching function m that is an essential part of the rule instance is derived by matching the left side to the genuine net. In a second step, replacements in graphs I and R are made according to replacements made in the first step and using given data and information stored in a lookup table. In Figure 15, a sample graph extracted from the rule skeleton (as shown in Figure 14), which is using both types of nonterminals can be seen. Before discussing the example, the replacement algorithm, which is capable of collecting the replacements for nonterminals and apply the replacements to the skeleton, will be described in more detail.

Replacement Algorithm (REP)

1) *Lookup table generation:* This table is generated from input parameters and values specified in the rule class. It can contain values of various types including nets. Keys are derived from the rule class and should match names of nonterminals in the skeleton. An empty or invalid lookup table is prevented by the rule class validation step executed before the replacement (Figure 13 ②).
2) *Lookup table replacement:* All nonterminals with matchings in the lookup table are replaced in the skeleton's nets.
3) *Net matching:* Based on the partially replaced skeleton and especially of the partially replaced net L, a possible matching in the genuine net is identified to
    a) *replace nonterminals* that have not been replaced using the lookup table and
    b) find entering points/nodes in the genuine net for a *net traversal*.

4) *Net traversal:* The genuine net is traversed according to the previously found matching(s) and due to the specified traversal method (as specified in the rule class), which will be described in more detail below.
5) *Net nonterminal replacement:* Net nonterminals are replaced by the previously derived subnet(s).
6) *Completion of lookup table:* The lookup table gets extended with the derived subnets from net matching and traversal bound to the nonterminal names used in the skeleton.
7) *Initiate redesign:* The redesign as defined in the rule class is initiated, such that all nonterminals are replaced in the redesign and the redesign is applied to the physical representation.

Net traversal is basically implemented by a simple algorithm. It traverses the genuine reference net simply by following the directed edges through the net from a pre-defined starting node (derived from a matching of the rule skeleton's nets to the genuine net). Guard conditions and further semantic information are ignored for the traversal because the traversal aims at extracting a certain sub-structure of the net without considering the token play during runtime. If the traversal algorithm hits a transition representing an operation node, the inscription is interpreted regarding the identification of the corresponding recall transition. Here, the relevant ID is extracted from the inscription and tried to be matched to other transitions' inscriptions. If the algorithm cannot find a matching transition, it terminates. A second termination condition of the traversal is that the algorithm hits a node in the net that has no further edges leaving it.

Net matching and net traversal are further used to derive the matching function m necessary for the final rule definition. Note that m specifies the subnet of the genuine net that is rewritten by the rule. Furthermore, the net matching is possibly not unique leading to 0 to n resulting rules from this step in the algorithm. In the case of $n > 1$ matchings, n rule instances result from the process (Figure 13).

Figure 15 shows the application of the REP algorithm onto the left side of the rule skeleton, as defined in Figure 14. Assumed that `WC` is predefined by given input data, such that `WC` is defined as the string `setA`, the initial lookup table is filled with this information. Therefore, in the first step `WC` is replaced by `setA` resulting in the first intermediate result. In the next step, the genuine net is matched to the intermediate result. Thereby, the nonterminal `VAR` can be replaced by `x`. Subsequently, the genuine net is traversed given the initial node, in this case the transition is inscribed with `NET_0 trav`. Finally, the result of the traversal is added to the skeleton.

The whole rule instance is derived by further application of the lookup table replacement step of the REP algorithm to all nonterminals. According to the algorithm's last step, the traversed subnets are part of the lookup table and thereby can also be replaced in I and L. Thus, the lookup table should contain all necessary replacement elements due to the rule class validation and the previously applied net traversal.

2) Boxing: Up to this point, only graph grammar and EBNF-like replacements of nonterminals applied to graphs were used. For more complex rules, for instance, rules involving a number of interaction elements that are not fixed in advance, further structures are necessary extending the current modeling approach of rule skeletons. Therefore, the basic skeleton description is extended by *boxes* that represent *loops* or *alternatives*, which are parameterized during runtime as specified in the rule class. Thus, from this boxed description of a rule skeleton, a simple nonterminal-based representation is derived algorithmically before the REP algorithm generates the final rule. This extraction algorithm is composed of recursive calls of loop box and alternative box extractions following the nesting of boxes of both types. Before discussing the extraction algorithm in more detail, the semantics of loop and alternative boxes will be discussed in detail.

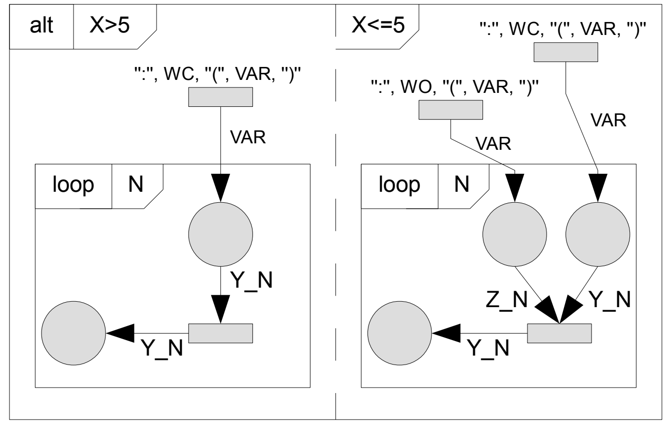


Figure 16. Example of using boxes in rule skeletons: a combination of alternative and loop boxes.

Loop box: A graph defined in a loop box gets replicated as many times as defined. Therefore, a loop box gets parameterized by the number of copies that should be created. Figure 17 shows a loop box indicated by the keyword `loop` followed by the parameter `N`. This parameter can also be used as nonterminal in the graph. In Figure 17, the nonterminal `WC_N` represents for a call method dedicated to an individual interaction element. The iteration counter `N` added to the nonterminal `WC` specifies that every created copy of the given net has to be matched to a method name from an individual interaction element. The matching has to be specified in the rule class and generated during the generation of the lookup table in the REP algorithm.

Alternative box: This box defines two different graphs to be selected in the extraction phase according to the specified condition. This condition is evaluated in an if-then-else fashion. If the condition is evaluated to true, the graph in the left box is selected for further extraction, otherwise the right graph gets selected.

Boxes can be used in a hierarchical fashion, such that boxes of different type can be nested into one another. Boxes can also be used in parallel, such that one alternative box holds a graph that uses two loops on the same hierarchical level. In general, it is possible to box subgraphs, as it can be seen in Figure 17. Regarding the use of boxes in the specification of rule skeletons, these boxes have to be extracted before the replacement algorithm can be applied to a plain rule skeleton. The following algorithm is applied to boxed rule skeletons to retrieve a plain rule skeleton for input into the replacement the REP algorithm. The extraction algorithm is mainly based on two steps: the first step resolves the nesting of the boxes and the second step applies the extraction to the boxed rule skeleton. The algorithm can be specified as follows:

Extraction Algorithm (EXT)

1) *Create Box Tree:* Starting on the highest level defining the root of the tree (which could be seen as the left, interface, or right side of the rule), all boxes on the next lower level are identified. For every box, one child will be created. This procedure is repeated until no more boxes can be identified on the next lower level (i.e., the lowest level has been reached in a subtree). If the currently selected node references an alternative box, the condition of the box gets evaluated before the children are inspected further, such that the selected subgraph can be inspected without creating unnecessary subtrees that would not be interpreted in the following step according to the evaluation of alternatives.
2) *Extraction:* To extract the final graph, the box tree is traversed in post-order. Each time a root node of a subtree gets selected by the traversal, the corresponding box is 'executed'. In case of an alternative box, nothing happens because it has already been evaluated in the box tree creation, before. In case of a loop box, the 'copy' operation is performed as often as specified. After finishing the box execution, the tree traversal process is continued until the box tree traversal ends in the root node.

In Figure 17, an example of an application of the extraction algorithm can be seen. To show the general process in all details, we decided not to discuss this example in context of the previous introduced simulation of a nuclear power plant (see Section III-C). The depicted rule stub is comprised of an alternative box and various loop boxes. Depending on the value `X`, one or two loop boxes have to be extracted. In the box tree creation step, the alternative box is inspected as the box on the highest level of the skeleton's box hierarchy. In this example, `X` is set to `3`. Thus, the right graph is selected for further inspection and a node is added to the box tree. Furthermore, the alternative box is removed from the rule skeleton as preprocessing of the extraction step. In the next step, the two loop boxes get inspected. For each box, a node is created in the tree each referencing one of the boxes. Next, the create box tree step terminates because no more boxes are encapsulated in the loop boxes.

The first extraction step as shown in Figure 17 starts with the traversal of the box tree (in post-order), first selecting the left loop box for extraction. The result of the first extraction step is shown on the lower left side of Figure 17. `N` is set to `2` resulting in two copies of the subgraph as defined in the rule skeleton. It can also be seen that the box crossing edges are duplicated in this case. The extraction of the second loop box is shown in the lower right corner of Figure 17, following the same extraction operation. The next node selected in the box node tree is the node representing the alternative node, which has been previously removed in the box tree creating step. The next and last node selected is the root node causing the EXT algorithm to terminate.

The EXT algorithm is applied to all graphs of the rule skeleton before the left side is inputted into the REP algorithm for nonterminal replacement. After finishing the EXT and REP algorithm, the rule has been generated. Before discussing a more complex example that implements adaptive automation using this approach, the rule class description has to be further specified, using XML as described below.

C. Rule Class Description

The rule class description, as briefly discussed above, specifies which data has to be provided to the REP and EXT algorithms to finally create the application specific rule. Beside the rule skeleton and the needed data, the rule class contains a description of necessary changes of the physical representation. At a glance, the following information is specified in a rule class:

- Metadata
- Selection heuristics and use of traversal algorithm
- Instantiation precondition
- Nonterminal declaration and definition
- Box parameter specification
- Rule skeleton
- Redesign of the physical representation

Metadata mainly specifies information regarding the sort of rule class, how it gets instantiated (interactively or system-side), its name, and some human readable description. The net traversal parameter specifies how the nets are traversed for rule skeleton instantiation. Currently, only the `standard` algorithm is implemented, as discussed above. The selection heuristic specifies how a certain rule is selected from a set of rules resulting from the instantiation of a rule skeleton. Currently, only the `first` strategy is available, simply selecting the first successfully generated rule. Further, a set of instantiation preconditions can be specified. Therefore, variables can be defined, which are used in these conditions in a second step. The variables are then matched against input data during runtime, similar to nonterminal symbols.

The declaration and definition of nonterminal symbols mainly specifies how values for the REP algorithm are derived or how certain nonterminals are associated to specific values. Values specifying box parameters are necessary to evaluate loop or alternative boxes in a rule skeleton. All specifications either define specific values (constants) or define how values will be derived; through user interaction or system side data input. Using concepts like RDF, the data specification in the rule class can be defined even more flexible. In Section V-B, this will be discussed in more detail with an example. Declaration of nonterminals and box parameters will be defined as shown in the XML snippet given in Appendix B.

In this context, RDF is used to specify nonterminal's datatype declarations and the specification of box parameters used in the associated rule skeleton. This makes an implementation independent description of rule classes possible, such that the concept is not restricted to be used with its initial implementation. For instance, the nonterminal `WO` (see

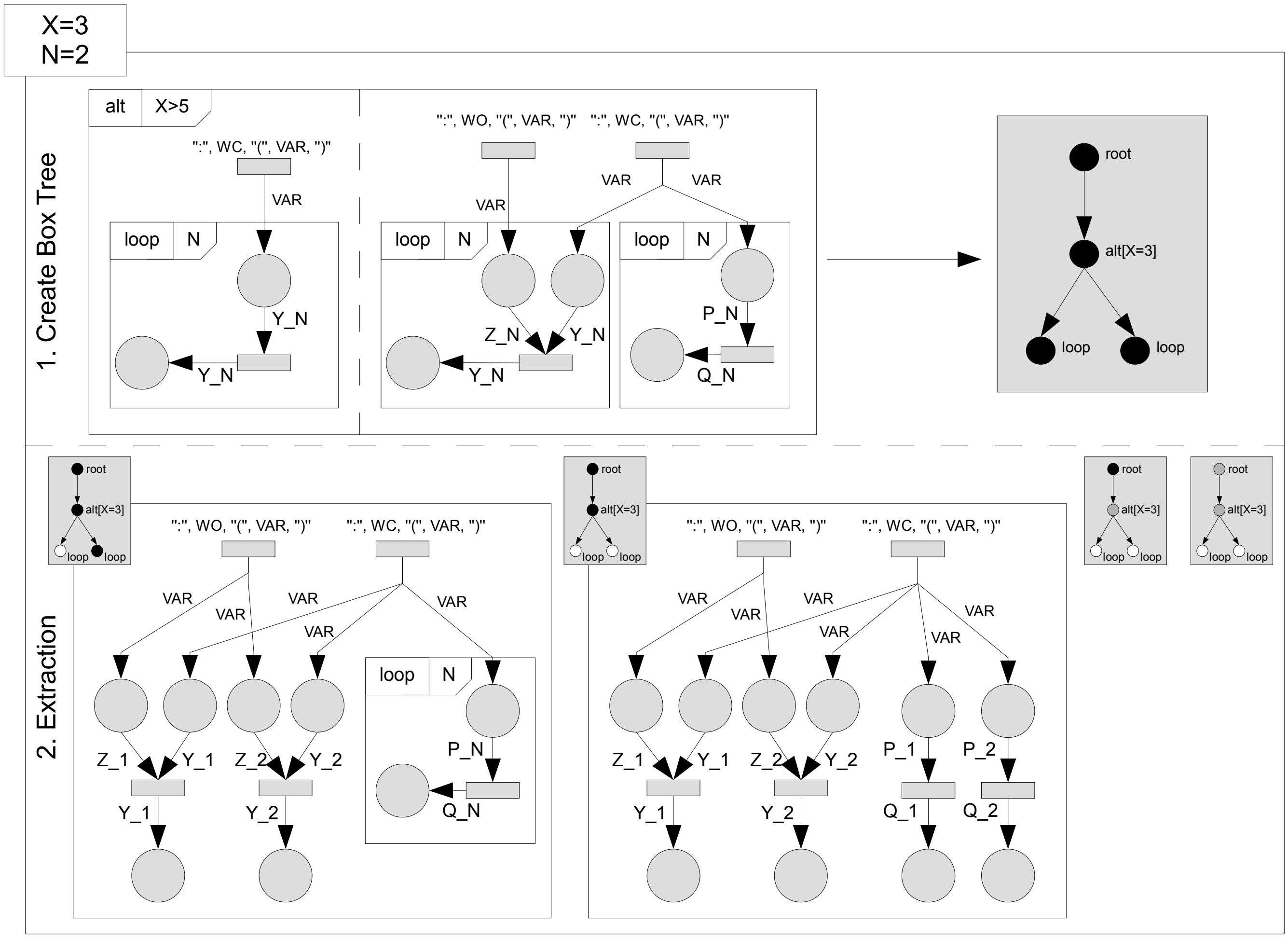


Figure 17. The application of the EXT algorithm, which extracts a boxed graph skeleton.

Appendix B) represents an interactively selected widget as can be identified by the specified datatype `widgetInteract`. `WC` represents a newly added widget, which needs an extension of the physical representation. This aspect will be further discussed below. Another example is the box parameter `X`, which is also inputted interactively as integer value (specified as `intInteract` datatype). Furthermore, a description has been specified to be shown to the user explaining what sort of parameter `X` is.

The generated rules are applied using the UIEditor implementation based on the DPO approach. This implementation uses a description of the DPO rule, where R, I, and L are defined as PNML-based graph specification. PNML (Petri Net Markup Language) is a markup language for specifying Petri nets of various types [48]. To stay consistent with this implementation, the description of rule skeletons will be based on this description concept simply extended by two types of XML nodes specifying loop or alternative boxes. The XML stub given in Appendix C shows how the rule skeleton in Figure 17 is expressed in the extended PNML format.

This extended net description is embedded into a rule description as discussed above (see Section III-D). Thus, three rule skeletons are specified with an additional specification of a mapping function.

In addition to changes of interaction logic, changes of the physical representation can occur, as it is the case for the nonterminal `WC` in Figure 14. In this case, a button should be added, which is connected to the interaction logic using the replacement of the `WC` nonterminal as reference. The REP algorithm is capable of creating a new interaction element as described in the rule class, adding it to the physical representation, and deriving the relevant reference from it to replace the relevant nonterminal in the rule skeleton. The redesign element for the rule class is described by the following XML snippet in Figure 18.

In this first version, only simple changes to the design of the physical representation are added: `newWidget` adds a specified widget to the physical representation and

```
<rc:redesign>
  <rc:newWidget reference="WC"
     widgetType=
       "http://uieditor.org/widgets/button"
     method=
       "http://uieditor.org/widgets/actionEvent"/>
  <rc:deleteWidget reference="WO"/>
</rc:redesign>
```

Figure 18. XML snippet of a redesign as specified in a rule class.

`deleteWidget` removes an existing widget from the physical representation. The interpretation system has to decide where to add the widget and in what initial size. In context of redesign, various extensions could be made, such as changing existing interaction elements regarding their outward appearance or their specific functionality. Still, this sort of change needs a more specific description concept of the physical representation and its redesign. The current version of the UIEditor uses a proprietary format for describing the physical representation. As future work, the use of UIML or UsiXML is planned to make the description of the physical representation more flexible and interchangeable between different (hardware) platforms.

Finally, an entire rule class is specified according to the example given in Appendix D. The following section will introduce and discuss an entire example using the rule generation process including the formalization of rule skeletons and rule classes based on XML as well as the application of the REP and EXT algorithms. The example presents a concept of adaptive automation as presented in [1]. It shows the usecase of a water pump as part of the nuclear power plant simulation scenario introduced above.

V. Adaptive Automation

Before presenting an example for adaptive automation in detail, the main motivation for implementing adaptive automation concepts will be discussed. The basis for this argumentation is that research in cognitive psychology has revealed important consequences of automation with respect to the human operator's workload in monitoring and control of technical processes, especially in critical, non-standard situations, as has been described in [49] and [50]. High workload is closely related to error rate, as well as to factors that influence the error rate in human-machine interaction, such as motivation, well-being, or situation awareness, as has been described in [51] and [52]. As has been discussed above, adaptive user interfaces are capable of reducing complexity in the interaction with technical systems [3]. Thus, it seems obvious to adapt user interfaces in order to suit particular users' needs and to introduce into the adaption process the degree of automation as an important parameter influencing human factors in human-machine interaction [53]. Here, the degree of automation defines whether the user has more or less control over the process, which system information in a critical situation is provided, or how the granularity of input operations is defined.

Therefore, adaptive automation will be discussed along the running example of a simplified nuclear power plant simulation. In this context, it is assumed that the mental workload of a reactor operator is measured for triggering and instantiating a user interface reconfiguration based on the rule generation concept discussed above and the UIEditor implementation with its associated formal user interface modeling approach. Especially in context of automated systems, the degree of automation is associated with a potential increase of mental workload and thereby is an indicator whether the degree of automation is too high or too low and whether it should be adapted or not. Weert [54] describes how mental workload can be measured based on different physiological factors, such as heartbeat rate, facial expression, perspiration, or eye blink rate. Out of these factors, pupillometry has been identified as a promising measurement tool for workload, especially in context of adaptive automation to increase human performance [55]. This gives an idea of how mental workload could be measured in the scenario presented above.

The section below discusses the use of the previously introduced approach for implementing a simple adaptive user interface, which is capable to adapt the degree of automation according to measured mental workload.

A. Example for Adaptive Automation

For making the degree of automation adaptable through a formal adaptive user interface model, it is assumed that the automation concept is fully accessible through an external formal model that matches the underlying concept of formal user interface modeling; therefore, the automation model should employ a reference net-based representation. If this assumption holds, the automation process can be also introduced into a rule skeleton and thereby can be introduced into the interaction logic through the rule generation and application process. Thereby, automation can be understood as formal abstraction of interaction processes between the human user and any given system that has been technically implemented. Thus, in our sense, automation is part of the interaction logic and simultaneously defines the degree of automation as visible from the user's perspective.

The automation of steering a water pump will be used as use case in the presentation, below. This use case of adaptive automation will be discussed according to a discrete and recurrent process of two operations: increase *(inc)* and decrease *(dec)* of the rounds per minute of a water pump. Here, it should be assumed that these operations have to be executed in an iterative fashion, such as the process shown in Figure 19. Thus, the process increases and decreases the rounds per minute (rpm) iteratively. According to the former assumption, this process can be introduced into reference net-based interaction logic as indicated by the bold arrow in Figure 19. The automation of this process can then be started by the user pressing the newly added "Start" button after the reconfiguration and redesign indicated by the bold arrow to the existing user interface. From this point on, the user is only able to monitor the system's state by observing the tachometer-like output widget, showing the pump's current rpm being controlled. Thus, using the reconfigured interface (indicated by **(I)** in Figure 19), she is not able to follow the operations that are automatically executed by the interaction logic.

As Parasuraman describes in [49] and [50] that workload increases during critical situations because the user has to understand the system's current situation, as well as how the automated control processes are reacting to the situation. The user has to gain insight into the automated process, resulting in an increase of mental workload, sometimes dramatically. This problem occurs in the example after the the first step of reconfiguration and redesign has been applied to the initial user interface. The user has no insights to the automated process, except that it is running. To adapt automation to this situation, the user interface **(I)** can be reconfigured (see Figure 19 **(II)**), by adding more interaction elements providing deeper insights into the automated process. Two lamps are added to

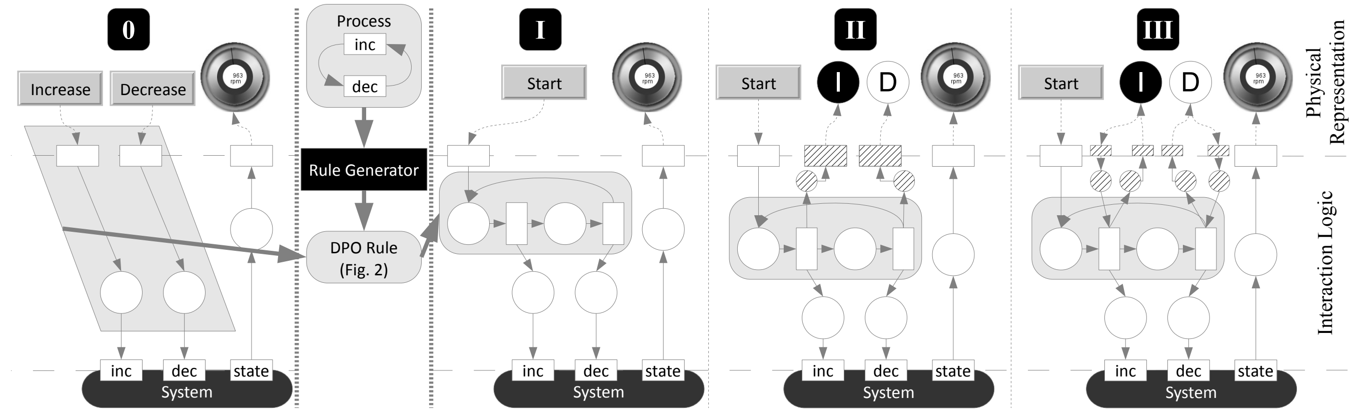


Figure 19. An example of how to change the degree of automation through reconfiguration of interaction logic, using rule generation.

the physical representation accompanied by an extension of the interaction logic, now showing which operation is executed at any given moment. Thus, the user is now able to see when the automation increases or decreases the speed of the pump. This makes interaction more finely grained and transparent to the user. A further reconfiguration extends the first by changing the simple lamps into buttons (see Figure 19 **(III)**), where the user is now able to control the automated process and thus gains more control over the still automated control process of the pump. Another possibility would be to remove the automation from the interaction logic and give all control back to the user without restriction or even, contrarily, to reduce the interaction and fully automate the process. The first would result in the initial user interface (see Figure 19 **(0)**).

B. Implementation using the Rule Generation Concept

The section above just discussed possible reconfigurations and redesigns that could be applied to a user interface for controlling the speed of a water pump in the nuclear reactor simulation. This example shows how the degree of automation could be increased or decreased using the reconfiguration concept introduced in this paper. Still, it has not been addressed how the rule generation process can implement this scenario. Therefore, this subsection discusses the reconfiguration step **(I)** of Figure 19. The needed data will be characterized for instantiating the rule class that will be described in a next step.

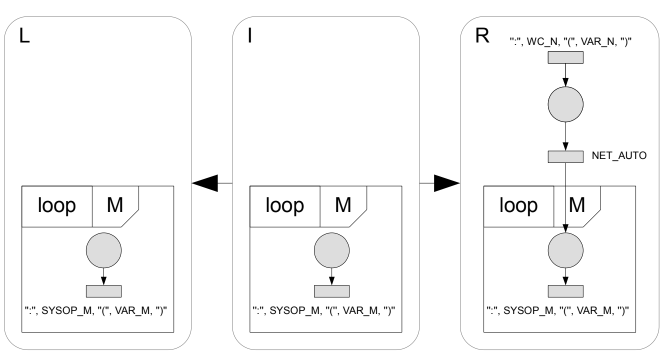


Figure 20. Rule skeleton for deriving the rule necessary for applying reconfiguration step **(I)** as defined in Figure 19.

As presented above, cognitive load is a relevant value to trigger an adaptation of the user interface according to its value. Various works have identified pupillometry as a possible indicator of user's workload, as these by Weert [54], de Greef et al. [55], and Halverson et al. [56]. Thus, the rule generator can be triggered by a component that measures workload through pupillometry. Here, it can be seen that the instantiation of a rule class is accompanied with a use case dependent pre-processing, in this case the mental workload. This is necessary to trigger the rule class instantiation process as defined in Figure 13. In the rule class validation step, the mental workload value is further used to decide, (a) which rule class should be used and (b) whether a selected rule class should be applied to the user interface. The XML snippet given in Appendix E shows one possible implementation of a rule class describing step **(I)**, which would be validated as a rule class that can be applied to the user interface.

In Figure 20, the rule skeleton used in this rule class can be seen, which is the relevant structure, which defines the reconfiguration applied to the user interface in step **(I)**. The class definition shows that the automation model has to be provided from outside as graph (cf. `externReferenceNet` in Appendix E), thus by the tool that triggers the reconfiguration. Furthermore, the system operations (*inc* and *dec* in the above example) as well as the involved interaction elements, which are removed from the physical representation have to be provided by this external tool. Still, the latter could also be provided by a net traversal that identifies the widget transitions connected to the specified system operations. Here, it was decided to reduce the complexity of the rule skeleton by assuming the interaction elements to be provided by the triggering instance. Finally, the subnet `NET_AUTO` has to specify which transition is connected to the newly added interaction element and which transitions are connected to the system operations *inc* and *dec*. This could also be done using the specification of rule skeletons, as has been discussed above. Here, through matching of inscriptions, the correct mapping can be algorithmically derived. The finally derived rule can be seen in Figure 19, which is comprised of the application of the REP and EXT algorithm as discussed above.

VI. Conclusion and Future Work

The paper at hand introduced a new approach to algorithmic rule generation as basis for flexible and formal creation of adaptive user interfaces. The whole approach is based on a formal modeling language called FILL that is algorithmically transformed into reference nets, a special type of Petri nets. This transformation equips FILL with formal semantics as well as making it executable. This formal modeling approach is used to describe interaction logic of a user interface, which is further extended by a proprietary XML-based format describing the physical representation of a user interface. By applying the DPO graph rewriting approach, this kind of user interface model becomes formally adaptable and thereby fulfills the requirement of a self-contained approach for formal modeling and reconfiguration of user interfaces, as has been defined in the beginning.

Nevertheless, the implementation or creation of adaptive user interfaces needs an algorithmic and computer based approach for a flexible creation of adaptation rules applied to a user interface. Therefore, we introduced a new rule generation concept based on an XML specification of rule classes, equipped with a formal description of rule-skeletons based on graph and string grammars. This makes a flexible declaration of rules possible, as has been shown in a concluding example. This example discusses the creation of an adaptive user interface for changing the degree of automation regarding a user interface to control the throughput of a water pump as part of a nuclear power plan simulation. Here, automation becomes a part of the interaction logic of a user interface.

Future work aims at extending the simply structured user interface modeling approach to be more modularized. This makes a separation of dialog and system models in the interaction logic possible. Furthermore, the presented approach will be completely implemented and investigated in an evaluation study. Here, mainly the aspect of adapting a user interface according to measured mental workload will be investigated in cooperation with cognitive psychologists in the context of a working environment. Questions concerning helpful adaptations and restrictions according to changes in the user interface will be the focus of our research. Finally, it is planned to further identify extensions to the rule class and rule skeleton descriptions following from requirements in other usecases rather than in context of adaptive automation.

Appendix

Appendix A - Below, an example for a DPO rule specification is given in its specific XML format, which is used for applying reconfiguration to reference net-based interaction logic. All nets of the rule are specified using PNML, the Petri Net Markup Language. `deleteNet` references the left side of a DPO rule, `interface` denotes the interface graph I of a DPO rule, where `insertNet` dedicates to the right side of a DPO rule. The DPO rewriting approach and the associated rule description concept is subject of discussion in Section III-D.

```
<rule>
  <deleteNet>
    <net>
      <place id="p1"/>
      <place id="p3"/>
      <transition id="t2">
        <inscription>
          <text>guard x==3;</text>
        </inscription>
      </transition>
      <arc id="a1" source="p1" target="t2">
        <inscription>
          <text>x</text>
        </inscription>
      </arc>
      <arc id="a2" source="t2" target="p3">
        <inscription>
          <text>x</text>
        </inscription>
      </arc>
    </net>
  </delecteNet>
  <interface>
    <net>
      <place id="p1"/>
      <transition id="t2">
        <inscription/>
      </transition>
    </net>
  </interface>
  <insertNet>
    <net>
      <place id="p1"/>
      ...
    </net>
  </insertNet>
  <mapping>
    <mapElement insertID="p1"
      interfaceID="p1" deleteID="p1"/>
    ...
  </mapping>
</rule>
```

Appendix B - Below, an example of a rule class specification is given as XML file, where `rc` specifies the namespace for rule classes and `bx` the namespace for boxes in rule skeletons. Rule skeletons are subject of discussion in Section IV-B.

```
<rc:ntdeclaration name="WO"
   rdf:datatype=
     "http://uieditor.org/nttypes/widgetInteract"/>
<rc:ntdeclaration name="WC"
   rdf:datatype=
     "http://uieditor.org/nttypes/newWidget"/>
<rc:ntdeclaration name="Y_N" iterate="N"
   rdf:datatype=
     "http://uieditor.org/nttypes/netTrav"/>
...
<bx:parameter name="N">
  <bx:value
     rdf:datatype=
       "http://uieditor.org/datatypes/int"
        value="2"/>
</bx:parameter>
<bx:parameter name="X">
  <bx:value
     rdf:datatype=
       "http://uieditor.org/datatypes/intInteract"/>
  <bx:description>
    This is a description shown in the interactive
    input box for this value.
  </bx:description>
</bx:parameter>
...
```

Appendix C - Below, an example of the extended PNML format for describing rule skeletons can be seen, where `pnml` denotes the namespace of PNML and bx the namespace for the box description. Nonterminals are specified as part of inscriptions related to transitions, places, or edges. These do not need an extension of PNML associated to a individual namespace. Rule skeletons are subject of discussion in Section IV-B.

```
<pnml:net>
  <bx:alt>
    <bx:if condition="X>5">
      <pnml:transition id="t1">
        <pnml:inscription>
          <text>":"WC,"(",VAR,")"</text>
        </pnml:inscription>
      </pnml:transition>
      <pnml:arc id="e1"
          source="t1" target"p1">
        <pnml:inscription>
          <text>VAR</text>
        </pnml:inscription>
      </pnml:arc>
      <bx:loop counter="N">
        <pnml:place id="p1"/>
        <pnml:place id="p2"/>
        <pnml:transition id="t2"/>
        <pnml:arc id="e2"
            source="p1" target"t2">
          <pnml:inscription>
            <text>Y_N</text>
          </pnml:inscription>
        </pnml:arc>
        <pnml:arc id="e3"
            source="t2" target"p2">
          <pnml:inscription>
            <text>Y_N</text>
          </pnml:inscription>
        </pnml:arc>
      </bx:loop>
    </bx:if>
    <bx:else>
      ...
      <bx:loop counter="N">
        ...
      </bx:loop>
      <bx:loop counter="N">
        ...
      </bx:loop>
    </bx:else>
  </bx:alt>
</pnml:net>
```

Appendix D - Below, an example of a complete rule class can be seen, which mainly specifies the structure of a rule class description. The introduction and discussion of rule classes can be found in Section IV-C. A concrete example of a rule class is given in Appendix E.

```
<rc:class
    xmlns:bx="http://uieditor.org/boxing/"
    xmlns:rc="http://uieditor.org/ruleClass/"
    xmlns:rule="http://uieditor.org/rule/"
    xmlns:pnml=
      "http://www.pnml.org/version-2009/grammar/"
    name="Interactive Widget Fusion">

  <!-- Class description>
  <rc:description>
    This class specifies the
    interactive fusion of n widgets.
  </rc:description>

  <!-- Selection Heuristic and Net Traversal>
  <rc:select type="first"/>
  <rc:travers type="standard"/>

  <!-- Instantiation precondition-->
  <rc:variable name="..." rdf:datatype="..."/>
  ...
  <rc:precondition con="..."/>
  ...

  <!-- NT and Box Parameter declaration -->
  <rc:ntdeclaration name="..."
      rdf:datatype="..."/>
  ...
  <bx:parameter name="N">
    ...
  </bx:parameter>
  ...

  <!-- Rule skeleton -->
  <rc:ruleSkeleton>
    <rule:deleteNet>
      <pnml:net>
        ...
      </pnml:net>
    </rule:delecteNet>
    <rule:interface>
      ...
    </rule:interface>
    <rule:insertNet>
      ...
    </rule:insertNet>
    <rule:mapping>
      ...
    </rule:mapping>
  </rc:ruleSkeleton>

  <!-- Redesign -->
  <rc:redesign>
    ...
  </rc:redesign>
</rc:class>
```

Appendix E - Below, an example of a rule class can be seen, that is dedicated to a concrete example of adaptive automation. This example is discussed in more detail in Section V-B.

```
<rc:class
    xmlns:bx="http://uieditor.org/boxing/"
    xmlns:rc="http://uieditor.org/ruleClass/"
    xmlns:rule="http://uieditor.org/rule/"
    xmlns:pnml=
      "http://www.pnml.org/version-2009/grammar/"
    name="Automate process">

  <!-- Class description>
  <rc:description>
    Increase automation
  </rc:description>

  <!-- Selection Heuristic and Net Traversal>
  <rc:select type="first"/>
  <rc:travers type="standard"/>

  <!-- Instantiation precondition-->
  <rc:variable name="MW"
    rdf:datatype="mentalWorkload"/>
  <rc:precondition con="MW<2"/>
  <rc:variable name="OP"
    rdf:datatype="int"/>
  <rc:precondition con="OP>=2"/>

  <!-- NT and Box Parameter declaration -->
  <bx:parameter name="M">
    <bx:value
```

```
      rdf:datatype="int"
      value="OP"/>
  </bx:parameter>
  <rc:ntdeclaration name="NET_AUTO"
      rdf:datatype="externReferneceNet"/>
  <rc:ntdeclaration name="SYSOP_M"
      rdf:datatype="sysOpName"/>
  <rc:ntdeclaration name="VAR_M"
      rdf:datatype="sysOp"/>
  <rc:ntdeclaration name="VAR"
      rdf:datatype="widgetInteract"/>
  <rc:ntdeclaration name="WC"
      rdf:datatype="newWidget"/>
  <rc:ntdeclaration name="WO_M"
      rdf:datatype="widgetInteract"/>

  <!-- Rule skeleton -->
  <rc:ruleSkeleton>
    <!-- See Fig. 18-->
  </rc:ruleSkeleton>

  <!-- Redesign -->
  <rc:redesign>
    <rc:newWidget reference="WC"
      widgetType=
        "http://uieditor.org/widgets/button"
      method=
        "http://uieditor.org/widgets/actionEvent"/>
    <rc:deleteWidget reference="WO_M"/>
  </rc:redesign>
</rc:class>
```

REFERENCES

[1] B. Weyers, “User-centric adaptive automation through formal reconfiguration of user interface models,” in CENTRIC 2013, The Sixth International Conference on Advances in Human oriented and Personalized Mechanisms, Technologies, and Services, 2013, pp. 104–107.

[2] J. T. Reason and J. T. Reason, Managing the risks of organizational accidents. Ashgate Aldershot, 1997, vol. 6.

[3] A. Jameson, “Adaptive interfaces and agents,” in Human-Computer Interaction Handbook. Erlbaum, 2003, pp. 305–330.

[4] K. Reinecke and A. Bernstein, “Improving performance, perceived usability, and aesthetics with culturally adaptive user interfaces,” ACM Transactions on Computer-Human Interaction (TOCHI), vol. 18, no. 2, 2011, pp. 1–29.

[5] B. Weyers, D. Burkolter, A. Kluge, and W. Luther, “Formal modeling and reconfiguration of user interfaces for reduction of human error in failure handling of complex systems,” International Journal of Human Computer Interaction, vol. 28, no. 10, 2012, pp. 646–665.

[6] B. Shneiderman, “Promoting universal usability with multi-layer interface design,” in ACM SIGCAPH Computers and the Physically Handicapped, no. 73-74. ACM, 2003, pp. 1–8.

[7] B. Weyers, Reconfiguration of User Interface Models for Monitoring and Control of Human-Computer Systems. Munich: Dr. Hut, 2012.

[8] P. A. Hancock, R. J. Jagacinski, R. Parasuraman, C. D. Wickens, G. F. Wilson, and D. B. Kaber, “Human-automation interaction research past, present, and future,” Ergonomics in Design: The Quarterly of Human Factors Applications, vol. 21, no. 2, 2013, pp. 9–14.

[9] F. Heidenreich, J. Johannes, M. Seifert, and C. Wende, “Closing the gap between modelling and java,” in Software Language Engineering. Springer, 2010, pp. 374–383.

[10] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks, “The semantic web: The roles of xml and rdf,” Internet Computing, IEEE, vol. 4, no. 5, 2000, pp. 63–73.

[11] A. Jameson, “Adaptive interfaces and agents,” Human-Computer Interaction: Design Issues, Solutions, and Applications, vol. 105, 2009.

[12] T. Lavie and J. Meyer, “Benefits and costs of adaptive user interfaces,” International Journal of Human-Computer Studies, vol. 68, no. 8, 2010, pp. 508–524.

[13] P. Langley and H. Hirsh, “User modeling in adaptive interfaces,” Courses and lectures-international centre for mechanical sciences, 1999, pp. 357–370.

[14] G. Fischer, “User modeling in human–computer interaction,” User modeling and user-adapted interaction, vol. 11, no. 1-2, 2001, pp. 65–86.

[15] S. Cheng and Y. Liu, “Eye-tracking based adaptive user interface: implicit human-computer interaction for preference indication,” Journal on Multimodal User Interfaces, vol. 5, no. 1-2, 2012, pp. 77–84.

[16] G. Kahl, L. Spassova, J. Schöning, S. Gehring, and A. Krüger, “Irl smartcart-a user-adaptive context-aware interface for shopping assistance,” in Proceedings of the 16th international conference on Intelligent user interfaces. ACM, 2011, pp. 359–362.

[17] R. Hervás and J. Bravo, “Towards the ubiquitous visualization: Adaptive user-interfaces based on the semantic web,” Interacting with Computers, vol. 23, no. 1, 2011, pp. 40–56.

[18] D. Navarre, P. Palanque, J.-F. Ladry, and E. Barboni, “Icos: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability,” ACM Transactions on Computer-Human Interaction (TOCHI), vol. 16, no. 4, 2009, pp. 1–18.

[19] O. Sy, R. Bastide, P. Palanque, D. Le, and D. Navarre, “Petshop: a case tool for the petri net based specification and prototyping of corba systems,” Petri Nets 2000, 2000, pp. 77–86.

[20] E. Barboni, C. Martinie, D. Navarre, P. Palanque, and M. Winckler, “Bridging the gap between a behavioural formal description technique and a user interface description language: Enhancing ico with a graphical user interface markup language,” Science of Computer Programming, vol. 86, 2013, pp. 3–29.

[21] Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, and V. López-Jaquero, “Usixml: A language supporting multi-path development of user interfaces,” in Engineering human computer interaction and interactive systems. Springer, 2005, pp. 200–220.

[22] M. Abrams, C. Phanouriou, A. L. Batongbacal, S. M. Williams, and J. E. Shuster, “Uiml: an appliance-independent xml user interface language,” Computer Networks, vol. 31, no. 11, 1999, pp. 1695–1708.

[23] F. de Rosis, S. Pizzutilo, and B. De Carolis, “Formal description and evaluation of user-adapted interfaces,” International Journal of Human-Computer Studies, vol. 49, no. 2, 1998, pp. 95–120.

[24] C. Janssen, A. Weisbecker, and J. Ziegler, “Generating user interfaces from data models and dialogue net specifications,” in Proceedings of the INTERACT’93 and CHI’93 conference on human factors in computing systems. ACM, 1993, pp. 418–423.

[25] G. Brat, C. Martinie, and P. Palanque, “V&v of lexical, syntactic and semantic properties for interactive systems through model checking of formal description of dialog,” in Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments. Springer, 2013, pp. 290–299.

[26] R. Bastide, D. Navarre, and P. Palanque, “A tool-supported design framework for safety critical interactive systems,” Interacting with Computers, vol. 15, no. 3, 2003, pp. 309–328.

[27] F. Paternò and C. Santoro, “Integrating model checking and hci tools to help designers verify user interface properties,” in Interactive Systems Design, Specification, and Verification. Springer, 2001, pp. 135–150.

[28] D. Navarre, P. Palanque, and S. Basnyat, “A formal approach for user interaction reconfiguration of safety critical interactive systems,” in Computer Safety, Reliability, and Security. Springer, 2008, pp. 373–386.

[29] D. Navarre, P. Palanque, J.-F. Ladry, and S. Basnyat, “An architecture and a formal description technique for the design and implementation of reconfigurable user interfaces,” in Interactive Systems. Design, Specification, and Verification. Springer, 2008, pp. 208–224.

[30] M. Blumendorf, G. Lehmann, and S. Albayrak, “Bridging models and systems at runtime to build adaptive user interfaces,” in Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems. ACM, 2010, pp. 9–18.

[31] J. Criado, C. Vicente Chicote, L. Iribarne, and N. Padilla, “A model-driven approach to graphical user interface runtime adaptation,” in Proceedings of the MODELS conference, IEEE. M. Jeusfeld c/o Redaktion Sun SITE, Informatik V, RWTH Aachen., 2010.

[32] O. Kummer, F. Wienberg, M. Duvigneau, J. Schumacher, M. Köhler, D. Moldt, H. Rölke, and R. Valk, “An extensible editor and simulation engine for petri nets: Renew,” in Applications and Theory of Petri Nets 2004. Springer, 2004, pp. 484–493.

[33] G. Abowd, R. Beale, A. Dix, and J. Finlay, Human-computer interaction. Prentice Hall, 1996.

[34] L. M. Reeves, J. Lai, J. A. Larson, S. Oviatt, T. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin et al., “Guidelines for multimodal user interface design,” Communications of the ACM, vol. 47, no. 1, 2004, pp. 57–59.

[35] S. A. White and D. Miers, BPMN Modeling and Reference Guide. Future Strategies Inc., 2008.

[36] O. Kummer, Referenznetze. Logos, 2009.

[37] B. Weyers and W. Luther, “Formal modeling and reconfiguration of user interfaces,” in Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the. IEEE, 2010, pp. 236–245.

[38] A. Schürr and B. Westfechtel, “Graph grammars and graph rewriting systems,” RWTH Aachen, Tech. Rep. AIB 92-15, 1992.

[39] H. Ehrig, R. Heckel, M. Korff, M. Löwe, L. Ribeiro, A. Wagner, and A. Corradini, “Algebraic approaches to graph transformation. Part II: Single pushout approach and comparison with double pushout approach,” in Handbook of graph grammars and computing by graph transformation, G. Rozenberg, Ed. World Scientific Publishing, 1997, ch. 4.

[40] H. Ehrig, K. Hoffmann, and J. Padberg, “Transformation of Petri nets,” Electronic Notes in Theoretical Computer Science, vol. 148, no. 1, 2006, pp. 151–172.

[41] H. Ehrig, K. Hoffmann, J. Padberg, C. Ermel, U. Prange, E. Biermann, and T. Modica, “Petri net transformation,” in Petri Net, Theory and Applications, V. Kordic, Ed. InTech Education and Publishing, 2008, ch. 1.

[42] J. Stückrath and B. Weyers, “Lattice-extended cpn rewriting for adaptable ui models,” in Proceedings of GT-VMT 2014 workshop, in press, 2014.

[43] B. C. Pierce, Basic category theory for computer scientists. MIT press, 1991.

[44] M. Heumüller, S. Joshi, B. König, and J. Stückrath, “Construction of pushout complements in the category of hypergraphs,” in Proceedings of the Workshop on Graph Computation Models, ser. GCM ’10, Enschede, The Netherlands, 2010.

[45] H. Ehrig and J. Padberg, “Graph grammars and Petri net transformations,” in Lectures on Concurrency and Petri Nets, ser. Lecture Notes of Computer Science, J. Desel, W. Reisig, and G. Rozenberg, Eds. Springer, 2004, vol. 3098, pp. 65–86.

[46] M. Weber and E. Kindler, “The petri net markup language,” in Petri Net Technology for Communication-Based Systems. Springer, 2003, pp. 124–144.

[47] H. Ehrig, G. Engels, and G. Rozenberg, Handbook of graph grammars and computing by graph transformation: Applications, Languages and Tools. world Scientific, 1999, vol. 2.

[48] J. Billington, S. Christensen, K. Van Hee, E. Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber, “The petri net markup language: Concepts, technology, and tools,” in Applications and Theory of Petri Nets 2003. Springer, 2003, pp. 483–505.

[49] R. Parasuraman, T. Sheridan, and C. Wickens, “A model for types and levels of human interaction with automation,” IEEE Transactions on Systems, Man, and Cybernetics: Systems and Humans, vol. 30, no. 3, 2000, pp. 286–297.

[50] R. Parasuraman and V. Riley, “Humans and automation: Use, misuse, disuse, abuse,” Human Factors, vol. 39, no. 2, 1997, pp. 230–253.

[51] C. Wickens and J. Hollands, Engineering psychology and human performance. Addison Wesley, 1999.

[52] M. Endsley, “Toward a theory of situation awareness in dynamic systems,” Human Factors, vol. 37, no. 1, 1995, pp. 32–64.

[53] R. Parasuraman, K. Cosenzo, and E. D. Visser, “Adaptive automation for human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and mental workload,” Military Psychology, vol. 21, no. 2, 2009, pp. 270–297.

[54] J. Weert, Ship operator workload assessment tool. Department of mathematics and computer science. Technical University Eindhoven, 2006.

[55] T. de Greef, H. Lafeber, H. van Oostendorp, and J. Lindenberg, “Eye movement as indicators of mental workload to trigger adaptive automation,” in Proc. of Augmented Cognition, HCII 2009, 2009, pp. 219–228.

[56] T. Halverson, J. Estepp, J. Christensen, and J. Monnin, “Classifying workload with eye movements in a complex task,” in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 56, no. 1. Sage Publications, 2012, pp. 168–172.

A Large-scale Power-saving Cloud System with a Distributed-management Scheme

Toshiaki Suzuki, Tomoyuki Iijima,
Isao Shimokawa, and Toshiaki Tarui
Central Research Laboratory
Hitachi, Ltd.
Yokohama, Kanagawa, Japan
{toshiaki.suzuki.cs, tomoyuki.iijima.fg,
isao.shimokawa.sd, toshiaki.tarui.my}@hitachi.com

Tomohiro Baba, Yasushi Kasugai,
and Akihiko Takase
Telecommunications & Network Systems Division
Hitachi, Ltd.
Kawasaki, Kanagawa, Japan
{tomohiro.baba.mn, yasushi.kasugai.rs,
akihiko.takase.wa}@hitachi.com

***Abstract*—A large-scale power-saving cloud system with a distributed management scheme is proposed. The system is composed of multiple data centers (DCs) connected by a wide-area network (WAN). In addition, it includes an inter-DC-management server, multiple DC-management servers, and multiple user VM-management servers. To reduce the power consumption of the DCs and the WAN, virtual machines (VMs) are migrated and data-routing paths are optimized under the condition that quality of service (QoS) is maintained by simultaneously providing necessary CPU resources and network bandwidth for services by the VMs. Aiming to enhance our previously proposed system, the management scheme is based on distributed management instead of central management. In the previous system, one DC-management server gathers all information to determine an appropriate alternative server to which VMs are migrated. On the contrary, in the proposed system, to distribute management load, each user VM-management server sends specifications of a VM to be migrated to other user VM-management servers. The other user VM-management servers then independently return a list of alternative servers that can accommodate the intended VM. After receiving the lists, the user VM-management server selects the most-suitable server. A prototype of the proposed system comprising 1,000 VMs, 400 servers, and four DCs was developed and evaluated. The time for determining reallocation of 1,000 VMs is within five minutes, which is about five times shorter than that taken by the previous system. These results indicate that the proposed system can reduce power consumption for one-week cloud operation by 30%.**

Keywords - power saving, QoS, cloud system, virtual-machine migration, distributed management, resource allocation.

I. INTRODUCTION

This work is an expansion of our previous work presented at ENERGY2013 [1]. In the previous work, a centralized management scheme for a large-scale power-saving cloud system composed of multiple data centers was focused on. In the current work, to enable the system to control a larger number of virtual machines (VMs), this scheme is extended to create a distributed management scheme. In addition, the performance of the extended scheme was evaluated in an environment with 1,000 VMs.

Lately, in conjunction with the increasing number of data centers (DCs) being constructed, the amount of electric power consumed by information and communication technology (ICT) systems has been dramatically rising [2]. As one of the biggest issues concerning ICT systems, including DCs, power-saving measures have therefore been attracting lots of attention [3].

To address power-consumption issues, many standardizations and technical developments aiming to make ICT systems more power efficient are being actively promoted. Although conventional activities have aimed at reducing the electric-power consumption of ICT systems, the respective power consumptions of the "server resource" and "network resource" are controlled separately. Conventionally, power-saving control has therefore been optimized on a resource-to-resource basis. However, total electric-power consumption by a whole large-scale cloud system, comprising multiple DCs and a wide-area network (WAN) connecting them, has not been optimized while service quality provided by the system is maintained. Besides, if power-saving control is conducted separately per resource, it might cause a serious problem for other resources. For example, an excessive aggregation of servers by VM migrations might degrade access quality to a VM since data flows are aggregated to the same routing path; as a result, network-link bandwidth is exceeded, and network congestion occurs.

With these issues in mind, we are aiming to develop an efficient power-saving control scheme for both network and server resources while "quality of service" (QoS) of networks and servers, such as bandwidth and CPU power, is guaranteed by integrated power-consumption management of both network and server resources. In a previous work [1][4], we proposed a power-saving cloud system centrally managed by one control server that gathers all information needed for determining allocation of VMs. This system, however, faces a scalability issue. In the present work, aiming at total power saving covering both WAN resources and DC resources, we propose a large-scale power-saving cloud system managed by cooperation between a WAN management server, user VM-management servers, and an inter-DC-management server.

The rest of this paper is organized as follows. Section II describes related works. Section III explains the requirements concerning a power-saving cloud system. Section IV proposes a large-scale power-saving cloud system with a distributed management scheme. The proposed system simultaneously saves electric power and guarantees access bandwidth to a VM. Sections V and VI respectively describe a prototype system and present the results of evaluations concerning power saving and determining reallocation of 1,000 VMs. Section VII concludes the paper.

II. RELATED WORK

To tackle power-consumption issues, many schemes have been proposed and standardization activities are ongoing. As for a power-saving scheme for ICT systems, "server-resource virtualization" (that is, saving power consumed by servers by optimizing necessary resources) has been under research and development [5][6]. In addition, power-saving schemes at the node and link levels [7][8] have been proposed. These schemes are useful for reducing the power consumption of our proposed cloud system at the link level. In addition, power-saving schemes [9][10][11][12] for the network level have been proposed. Power-saving schemes at the DC/server level [13][14][15][16] and at the inter-DC level [17][18] have also been proposed.

In the meantime, standardization activities, such as those undertaken by the Energy Management Working Group (EMAN) in the Internet Engineering Task Force (IETF) [19], the Institute of Electrical and Electronics Engineers (IEEE) [20], the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) [21], and the Distributed Management Task Force, Inc. (DMTF) [22], are continuing.

In conventional power-saving schemes like those mentioned above, network and DC/server resources are controlled separately and/or without consideration for the QoS of networks. In the current study, therefore, integrated management for maintaining network QoS and reducing energy consumption of servers is addressed.

III. REQUIREMENTS OF A POWER-SAVING CLOUD SYSTEM

A power-saving cloud system provides various services and resources, such as application software, CPU processing power, and storage, via a network. To create a power-saving cloud system and to reduce its total electric-power consumption during off-peak hours (such as late evening), only the minimum resources required for providing cloud services should be activated.

To control the power consumption of a target system, average loads on physical servers, VMs on those servers, and network nodes should be monitored in real time. In addition, VMs should be appropriately reallocated according to predicted future loads on servers and VMs when the load is under a predefined threshold during off-peak hours. After the appropriate reallocation of VMs, unnecessary physical servers should be turned off. The nodes or ports on the nodes that transmitted data to unnecessary physical servers should also be turned off or switched from active mode to sleep mode. Furthermore, service quality (such as access bandwidth to a VM) should be guaranteed before, as well as after, the power-saving control by VM migration. In addition, a power-saving scheme should be applied to not only small cloud systems comprising a single DC but also large-scale systems comprising multiple DCs.

To satisfy the above-mentioned requirements, the power-saving control should be executed according to the following procedures, namely, four power-saving policies.

- Policy 1: Power consumption of the DC can be reduced by turning off unnecessary physical servers that are no longer used after an appropriate reallocation of VMs in the DC by VM migration.
- Policy 2: Power consumption of the DC can be reduced by turning off unnecessary physical servers and network nodes that are no longer used after aggregation of running physical servers and data-transmission routes in the DC by VM migration.
- Policy 3: Power consumption of the DC can be reduced by turning off unnecessary physical servers and nodes in the DCs that are no longer used after aggregation of running physical servers and data-transmission routes by VM migration between DCs (based on cooperation between DC management and WAN management).
- Policy 4: Power consumptions of the DC and WAN can be reduced by turning off nodes (or their ports) in the WAN that are no longer used after VM migration between DCs and aggregation of data-transmission routes.

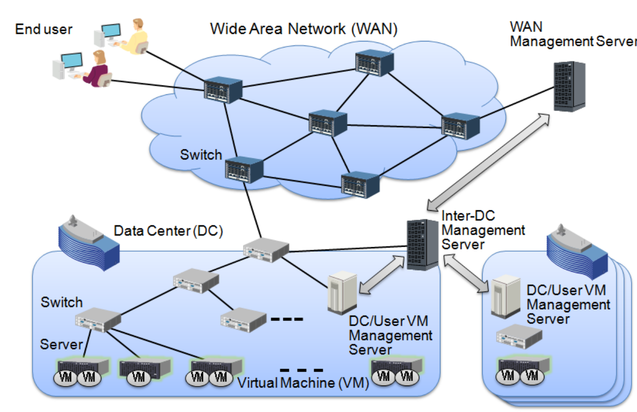


Figure 1. Proposed power-saving cloud system

On the basis of the above four policies, resources can be controlled from the viewpoint of power saving as well as from the viewpoint of service quality. More specifically, power consumption of the system should be reduced by aggregation of both server resources and network resources while service quality of a network path between an end user and the VM providing application services is maintained.

IV. PROPOSED POWER-SAVING CLOUD SYSTEM

A large-scale power-saving cloud system with a distributed-management scheme is proposed as follows. Specifically, a distributed power-saving management structure and its procedures are described.

A. System architecture

The architecture of the proposed power-saving cloud system is shown schematically in Figure 1. The system is composed of multiple DCs connected by a WAN. More specifically, the DC consists of multiple switches (SWs) for transmitting data, servers for providing various services, a

DC/user-management server for controlling user resources in the DC, and an inter-DC-management server for controlling multiple DC-management servers. The WAN consists of multiple SWs and a WAN-management server for monitoring and controlling resources in the WAN.

In the power-saving cloud system, the DC-management server monitors the loads of networks in the DC in real time. On the other hand, the user VM-management server monitors the loads of servers and VMs in the DC in real time. In addition, the DC- and user VM-management servers predict future loads of the resources, namely, servers, VMs, and SWs, by using statistical analysis (for example, by an autoregressive model [23]) based on the past history of loads. Specifically, the loads of these resources for eight hours are predicted according to their seven-day history of every five minutes. Besides, to reduce power consumption on the DC side, the management servers determine and control reallocation of resources such as VMs and routing paths.

To reduce power consumption of the WAN, the WAN-management server monitors loads of SWs in the WAN. It then gathers statistical-monitoring data and predicts future loads on each SW. Electric power consumed by the WAN is saved by optimizing data-routing paths and turning off SWs (or their ports) that are no longer used.

In summary, power consumption of the whole system is reduced by reallocating VMs between the DCs appropriately on the basis of cooperation between multiple DCs and user VM-management servers and the WAN management server.

B. *Distributed management structure*

The management structure of the proposed large-scale power-saving cloud system is shown in Figure 2. In this structure, one inter-DC-management server controls multiple DC-management servers. Each DC has one DC-management server that monitors network conditions of each DC and predicts future loads of the network. Multiple user VM-management servers monitor loads of multiple user servers and predict future loads of the servers and VMs. The user is assigned resources such as VMs and networks, and those resources are separated by a VLAN from other users' resources.

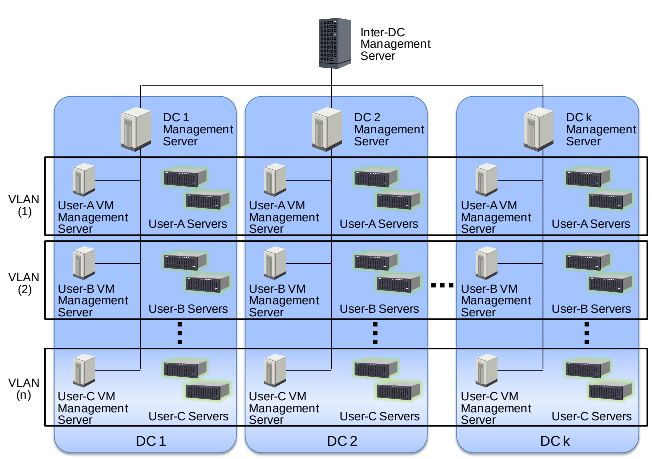


Figure 2. Distributed management structure for multiple DCs

C. *Overview of power-saving scheme by VM migration*

The process steps of a typical power-saving scheme based on VM migration by multiple management servers are shown schematically in Figure 3. In the proposed system, the inter-DC management server activates power-saving control according to the loads on the physical servers and VMs [step (1)]. The user VM-management server determines the order of VM migration [step (2)]. It then obtains "congestion potential" via the inter-DC-management server and the WAN-management server [step (3)]. To migrate VMs between DCs, the user VM-management server sends resource sizes, which are needed by the VM to be migrated, to other user VM-management servers in other DCs [step (4)]. These other user VM-management servers then send lists of servers that can accommodate the VM to be migrated [step (5)]. In addition, the original user VM-management server receives predicted future loads (e.g., SWs) on the DC network [step (6)]. To move the VM according to the lists received from the outside servers, the predicted loads of the DCs, and the effectiveness of the power saving, the user VM-management server determines one alternative server to which the VM is migrated [step (7)]. It then triggers an actual VM migration [step (8)]. The VM-migration result (i.e., notification of VM-migration completion) is transmitted from the user VM-management server to the inter-DC-management server [step (9)].

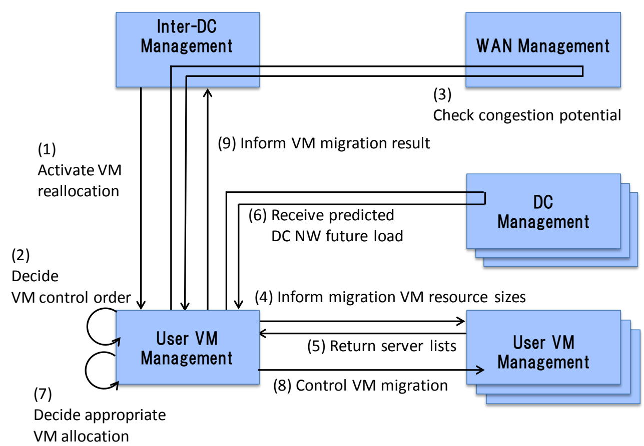


Figure 3. Process steps of reallocation of VM resources

D. *Detailed VM-resource reallocation*

The nine above-mentioned steps for resource reallocation are explained in detail as follows:

1) VM-reallocation trigger by inter-DC-management server: The inter-DC-management server starts (or stops) optimizing reallocation of VMs to each user VM-management server when the loads on servers and VMs are low (such as in the late evening).

2) Determination of VM-reallocation order by user VM-managementt server: The user VM-management server determines the order for reallocating running VMs in each virtual local-area network (VLAN). The reallocation order is determined according to (i) decending order of idle power,

(ii) ascending order of number of running VMs on a server, (iii) ascending order of assigned CPU resources, and (iv) ascending order of assigned memory resources.

3) Checking of congestion potential for the WAN by user VM-management server: To maintain access quality to a VM after the VM is migrated to another DC, the user VM-management server receives the congestion potential concerning the WAN from the WAN-management server via the inter-DC-management server. The congestion potential is evaluated on the basis of the history of the monitored data and predicted future loads in the case of fluctuation of bandwidth for each port of the switches. If network congestion is possible in the future, data-routing paths including the congestion point are not used for VM migration. More specifically, the IP address of the VM to reallocate, the identifier of the source DC, and the identifier of the VLAN to which the VM belongs are transmitted from the user VM-management server to the inter-DC-management server. A list of alternative DCs that can accommodate the migrated VM and the above-mentioned information from the user VM-management server are then transmitted from the inter-DC-management server to the WAN-management server. The congestion potential for the routing path between the user and an altenative DC is sent from the WAN-management server to the user VM-management server.

4) Informing other user VM-management servers about resource sizes of migrated VMs: The user VM-management server predicts future loads on the CPU and consumption of the bandwidth resource by the intended VM. It then sends the required sizes of resources by the VM to other user VM-management servers. In our previous system [1], the user VM-management server receives information concerning all servers from other user VM-management servers in other DCs. However, in that case, the user VM-management server has to choose one target server by itself from huge lists of alternative servers. However, scalability of this procedure is an issue. In the case of the proposed system, the user VM-management server therefore informs other user VM-management servers about sizes of required resouces to accommodate a VM. The other user VM-management servers then search for altenative servers.

5) Returning server lists of alternative servers: The user VM-management servers in other DCs receive the sizes of required resources to accommodate the VM to be migrated and search for altenative servers according to the required resources. The user VM-management servers in other DCs return lists of alternative servers that can accommodate the VM to be migrated.

6) Receiving predicted future load of a DC network: The DC-management server in each DC monitors the conditions of the network in the DC and predicts future loads of the network. The user VM-management server receives the predicted future loads of networks in each DC.

7) Determination of target server: The user VM-management server detemines an appropriate VM reallocation by considering all alternative DCs. Specifically, all servers that can provide enough resources to run the intended VM in the future and maintain access qualtiy to the VM at the same time are selected as alternative servers for the reallocation of the VM. The most-effective server for power saving is then selected as the final target server for the VM migration.

The user VM-management server receives lists of altenative servers that can accommodate the intended VM [steps (4) and (5)]. It also receives future loads of networks of other DCs [step (6)]. On the other hand, it predicts future loads on the CPU and consumption of the bandwidth resource by the intended VM [step (4)]. It temporarily determines several target servers to which the intended VM is reallocated by comparing the received available future resources for all alternative servers in other DCs and the amount of necessary resources for the intended VM.

The user VM-management server determines whether switches on the routing path between the entrance of the DC and an alternative server in another DC can provide enough bandwidth for the intended VM after the VM migration according to the infromation from step (3). On the basis of the monitored information from the WAN-management server, it checks the congestion potential for the routing path between the WAN edge connecting the DC and another WAN edge connecting an end user. To determine the most-appropriate alternative server, the user VM-management server checks all the above-mentioned evaluation points, i.e., CPU load, network congestion, and bandwidth. The most-appropriate server that can meet the requirements stated in Section III and has the most-effective power-saving advantage is then selected by the user VM-management server as the target server for the VM migration.

8) VM migration by user VM-management server: The VM migration is executed according to the trigger by the user VM-management server. As for VM-migration methods, various technologies have been developed [5], [6] and can be used for an alternative VM-migration scheme by combining them with the proposed power-saving cloud system. After executing the VM migration, the user VM-management server updates stored topology information. In addition, to predict future load, when the VM has been migrated to a server in another DC, the history of the VM’s resources (such as CPU load) is moved to another user VM-management server.

9) Information about VM-migration completion sent from user VM-management server to inter-DC-management server: After all VM migrations have been executed, the user VM-management server informs the inter-DC-management server that all VM reallocations are complete. In addition, the histories of migration from the source servers to destination servers are transmitted from the user VM-management server to the inter-DC-management server, which receives the migration histories and stores them. These histories are used when the migrated VMs are returned to the original allocated servers when CPU load increases.

E. Procedure for selecting alternative servers

The procedure for selecting alternative servers is depicted in Figure 4. The user VM-management server that is instructed to start VM reallocation by the inter-DC-management server takes the main role of a server for determining the allocation of VMs for each user. On the other hand, the user VM-management servers that belong to other DCs take the role of sub-servers.

The main user VM-management server makes a list that includes all servers that provide application services in order of power-consumption efficiency. In addition, it monitors loads of the VMs, servers, and their network ports. It then predicts future loads of those resources and manages the list with predicted loads of those resources. After receiving a request for reallocation of VMs, it sends predicted loads of the VM to be migrated to other user VM-management servers in parallel [step (1) in Figure 4].

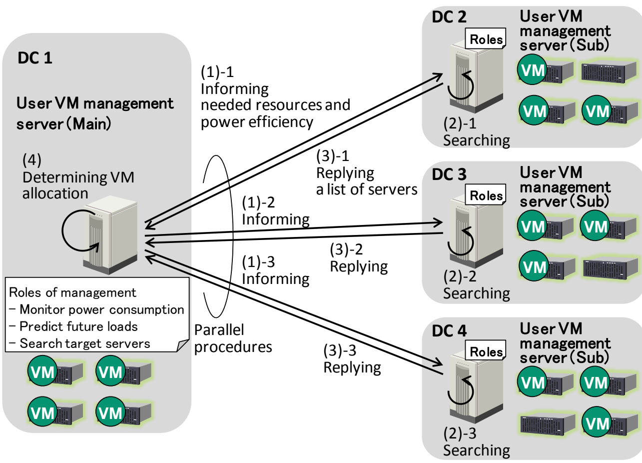


Figure 4. Procedure for selecting alternative servers

The other user VM-management servers acting as sub servers also make a list of servers to provide application services and manage it on the basis of the order of power-consumption efficiency. In addition, the sub user VM-management servers monitor loads of resources, such as the servers and their network ports, and predict future loads of those resources. When they receive requests to find suitable servers to accommodate the VM to be migrated from other DCs, each sub user VM-management server searches for possible servers by comparing the resources to be consumed by the migrated VM and remaining resources of each server [step (2)]. After finding possible servers, each sub user VM-management server sends a list of possible servers in parallel. In the process of finding the possible servers, when the other user VM-management servers find several alternative servers that can accommodates the migrated VM, they stop the search process and send the search result to the main user VM-management server that sent the request [step (3)].

When the main user VM-management server receives the lists of alternative servers from sub user VM-management servers, it selects several alternative servers in order of power-consumption efficiency. It then checks whether the alternative servers can accommodate the VM to be migrated and whether there is enough network bandwidth from a user to the DC to which the alternative server belongs. In addition, it determines one target server to which the VM is to be migrated [step (4)].

F. Scale-out function

When the loads of VMs' resources (such as CPU and consumed bandwidth) decrease, since the loads of resources used by each VM are very low, those VMs are aggregated to other servers to make "unnecessary servers". In addition, unnecessary servers are shut down or turned to sleep mode, and power consumption is decreased. On the contrary, when the loads of VMs or used bandwidth increase, the VMs should be distributed to multiple servers or the traffic should be detoured to other routes. When the VMs are again migrated, if the original serves are not on, they are turned on. To do that, the original server position where the VM is executed is memorized, and VMs are again migrated to the original servers when their loads increase.

G. Power-consumption model

A power-consumption model for the proposed cloud system is defined as follows. The amount of power (P_{All}) consumed by the cloud system is given by formula (1), where P_{IT} means power consumption of IT equipment, and P_{NET} means power consumption of network nodes. Formula (2) indicates P_{IT} is calculated by summing the power consumption of each server (P_{SV}) since the proposed system includes multiple servers as IT equipment. Here, i (i = 1, 2, 3, . . . , N) means the number of the server. In addition, n means CPU load (%) on the server. P_{SV} is given by formula (3) [24][25]. $P_{idle(i)}$ means the power consumption of the ith server during idle time, and $P_{max(i)}$ means power consumption under maximum load. Formula (4) gives P_{NET} of a network calculated by summing the power consumption of each node. Here, k (k = 1, 2, 3, . . . , M) means the number of the node. In addition, m means load (%) on a node in terms of bandwidth. The power consumption of the node (P_{NODE}) is given by formula (5) [7]. $P_{idle(k)}$ means power consumption by the kth node during idle time, and $P_{max(k)}$ means power consumption under maximum load. Here, P_{SV} and P_{NODE} are assumed to fit a linear function, as shown in Figure 5. The relations between power consumption and CPU load and between power consumption and traffic are independently evaluated in advance. According to that evaluation, the relation between power consumption and load (traffic) fits a linear function well (as shown in Figure 5).

$$P_{All} = P_{IT} + P_{NET} \quad (1)$$

$$P_{IT} = \Sigma_i P_{SV(i)}[n] \quad (2)$$

$$P_{SV(i)}[n] = P_{idle(i)} + (P_{max(i)} - P_{idle(i)})(n/100) \quad (3)$$

$$P_{NET} = \Sigma_k P_{NODE(k)}[m] \quad (4)$$

$$P_{NODE(k)}[m] = P_{idle(k)} + (P_{max(k)} - P_{idle(k)})(m/100) \quad (5)$$

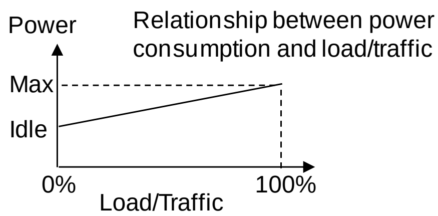


Figure 5. Assumed power consumption based on load/traffic

V. Evaluation of Power Saving

The performance of the proposed system with the centralized management scheme was evaluated by using an assumed CPU load model of a VM and consumed bandwidth by the VM per day.

A. *Evaluation system*

The evaluation system is shown schematically in Figure 6, and the number of pieces of ICT equipment is listed in Table I. In this system, switches, servers, and VMs in the DCs are emulated by open-source software, while switches in the WAN and management servers are real hardware. The performance (i.e., power-consumption reduction) of the power-saving control scheme for the DCs and WAN was evaluated as explained in detail in the following sections.

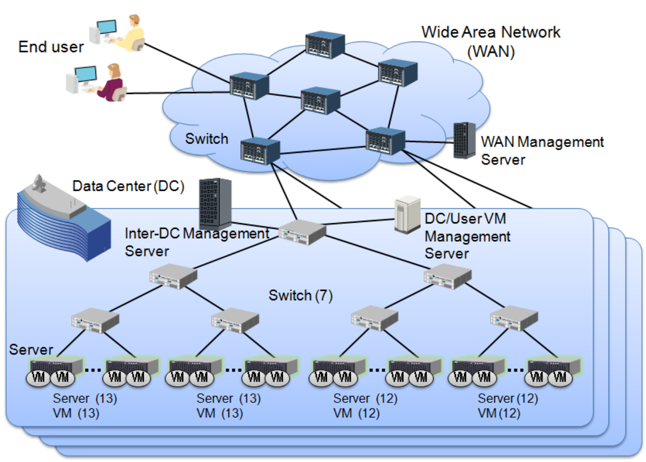


Figure 6. Evaluation system

TABLE I. Number of pieces of ICT equipment

	Item	Number of pieces of ICT equipment in four DCs
1	WAN-management server	1
2	SWs in WAN	6
3	DCs	4
4	Inter-DC-management server	1
5	DC/user VM-management servers	4
6	SWs in DCs	28
7	Servers in DCs	200
8	VMs on servers	200

B. *Evaluation of power-saving control for DCs*

The effectiveness of applying the power-saving control scheme for DCs per day was evaluated. First, a CPU-load model of a VM in the DC is assumed. The bandwidth consumed by the VM for one day is also assumed. The power consumed by DCs for one day is then evaluated on the basis of these assumptions.

1) *Workload model for a VM per day*

The assumed loads on the CPU as well as the incoming data flow to and the outgoing data flow from a VM are schematically shown in Figure 7. As depicted in the figure, the peak load is set only one time (around noon), and the loads during business hours are high, while the loads during the night (namely, those of the CPU, incoming flow, and outgoing flow) are low. The effectiveness of the power-saving control scheme is evaluated by comparing two cases: either executing appropriate VM reallocations or not.

The topology of the DC is shown in the lower part of Figure 6. The specifications and number of pieces of each apparatus in the DC are listed in Table II.

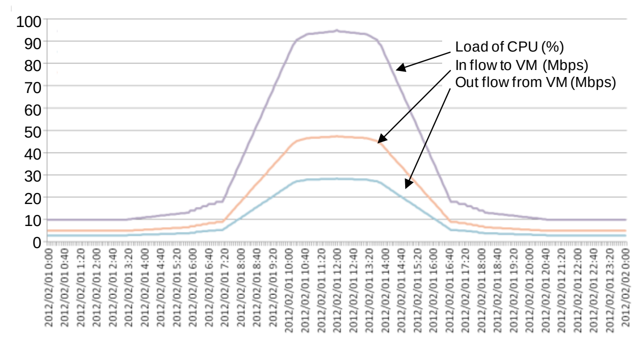


Figure 7. Load model of VM (CPU and in/out data flow)

TABLE II. Specifications and Number of Pieces of Each Apparatus In one DC

	Apparatus	Idle power	Max. power	Number
1	Server (Model 1)	120 W	170 W	17
2	Server (Model 2)	110 W	150 W	17
3	Server (Model 3)	177 W	251 W	16
4	VM	—	—	50
5	Switch	350 W	450 W	7

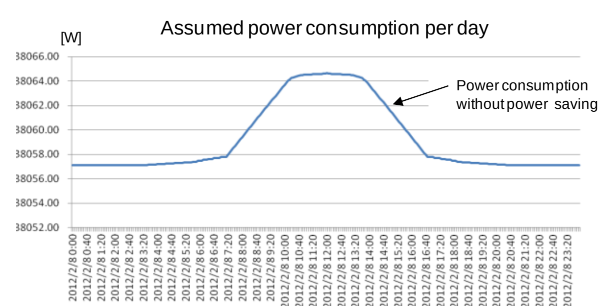


Figure 8. Electric-power consumption of a DC per day

2) Electric-power consumption of a DC per day

Electric-power consumption of a DC for one day (under the assumed loads for each server shown in Figure 7) is shown in Figure 8. The effectiveness of the power-saving control scheme under the following three conditions was evaluated. In the first condition, the VM is reallocated when the CPU loads are less than 75%. In the second and third conditions, reallocations are executed under CPU loads of 50% and 25%, respectively. On the other hand, when the load on the CPU is over these thresholds, reallocated VMs are returned to the original locations to maintain service quality.

3) Energy consumption of DCs per day

The evaluated fluctuations of power consumption of all DCs for the three above-mentioned conditions concerning power-saving control (CPU loads of 75%, 50%, and 25%) are shown in Figure 9. The result in the case of no VM reallocation is also shown in the figure for comparison. The figure verifies the effectiveness of the power-saving control scheme under the three conditions.

In addition, the results for VM reallocation to maintain VM access quality and energy consumption per day are listed in Table III. The number of VMs is shown in the upper row, while the number of servers (SVs) is shown in parentheses in the lower row. According to the table, some VMs are migrated between DCs (since the number of VMs in the DC changes after appropriate VM reallocations). In addition, the number of running servers is dramatically reduced after the VM migration. The CPU resource for a server is assumed to be enough for six VMs with CPU loads of 50%. The reductions in energy consumption under CPU loads of 25%, 50%, and 75% are 45.2%, 45.7%, and 47.6%, respectively.

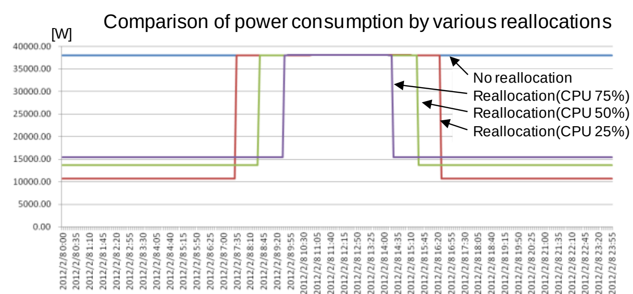


Figure 9. Electric-power consumption of a DC per day

TABLE III. ELECTRIC-ENERGY CONSUMPTION OF DCS PER DAY

	Optimization timing	DC1 VM (SV)	DC2 VM (SV)	DC3 VM (SV)	DC4 VM (SV)	Energy consumption per day (kWh)
1	No optimization	50 (50)	50 (50)	50 (50)	50 (50)	913.421
2	CPU load: 25%	60 (5)	48 (4)	48 (4)	44 (4)	500.388
3	CPU load: 50%	54 (9)	48 (8)	48 (8)	50 (9)	496.395
4	CPU load: 75%	52 (13)	48 (12)	52 (13)	48 (12)	478.323

C. Evaluation of power-saving control for a WAN

Power-saving control for a wide-area network (WAN) for one day was evaluated. In particular, the effectiveness of the power-saving scheme (based on bandwidth control by link aggregation) was evaluated. The topology of the evaluated WAN is shown in Figure 6. The specifications of the switches in the WAN are the same as those listed in Table II.

Power-saving control by appropriate data routing (including link-aggregation control) was executed after appropriate VM reallocation between DCs. The fluctuation of power consumption of the WAN is shown in Figure 10. Energy consumptions under the three types of control are compared in Table IV. According to these results, the reductions in energy consumption achieved by the power-saving control scheme under CPU loads of 25%, 50%, and 75% are 10.4%, 12.0%, and 13.7%, respectively.

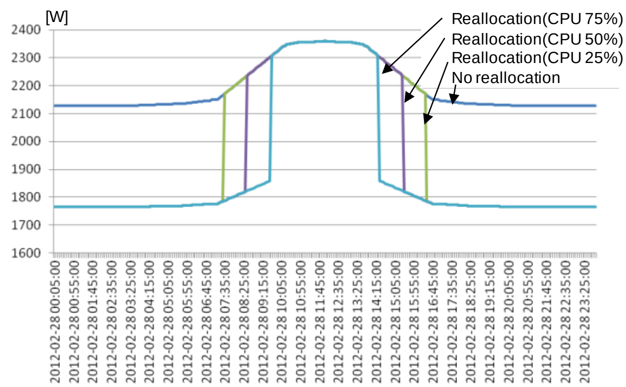


Figure 10. Electric power consumption of a WAN per day

TABLE IV. ELECTRIC-ENERGY CONSUMPTION OF ONE WAN PER DAY

	Optimization timing	Electric-energy consumption per day (kWh)	Reduction (%)
1	No optimization	52.470	–
2	25% CPU load	46.989	10.4
3	50% CPU load	46.194	12.0
4	75% CPU load	45.265	13.7

D. Power-saving effect for entire cloud system

The effectiveness of the power-saving control scheme for the entire proposed cloud system is shown in Table V. According to the table, the reductions of energy consumption achieved by the power-saving control scheme under CPU loads of 25%, 50%, and 75% are 43.3%, 43.8%, and 45.8%, respectively. In other words, energy consumption is reduced by approximately 40%. In addition, the highest reduction is accomplished under CPU load of 75%.

TABLE V. ELECTRIC-ENERGY CONSUMPTION OF ENTIRE CLOUD SYSTEM PER DAY

	Optimization timing	Optimization term	Electric-energy consumption (kWh)	Reduction (%)
1	No optimization	–	965.891	–
2	CPU load: 25%	15h00m	547.377	43.3
3	CPU load: 50%	17h00m	542.589	43.8
4	CPU load: 75%	19h10m	523.588	45.8

E. Discussion of power-saving effect

According to the results of this evaluation of a large-scale power-saving cloud system composed of multiple DCs and a WAN, energy consumption of the entire system is reduced by about 40% by the proposed power-saving control scheme. With regard to the power saving for the DCs only, energy consumption is reduced by over 45%. On the other hand, energy consumption of the WAN is reduced by only about 10%. The reason that the reduction of energy consumption of the DCs is high is the effectiveness of turning off unnecessary servers after appropriate VM reallocation. On the other hand, the reason that the reduction of the energy consumption of the WAN is low is that unnecessary switches were not turned off (since turning off unnecessary links is only possible under the assumed evaluation conditions). In the evaluation, migrations of management servers are not considered since they are not removable. Although energy consumed by them should be considered, their energy consumption is a bit small since the number of turned-off servers is much larger than that of management servers. In addition, the function for controlling network and server resources in a coordinated manner can be evaluated regardless of their energy consumptions. With regard to power saving for the entire proposed system, the reductions in energy consumption achieved by the power-saving control scheme under CPU loads lower than 25%, 50%, and 75% are 43.3%, 43.8%, and 45.8%, respectively. On the other hand, the times taken for the resource optimization under the three above conditions are 15 hours, 17 hours, and 19 hours and 10 minutes, respectively. When the power-saving control is executed under a CPU load of 75%, the time taken for the optimization is the longest, and reduction in energy consumption is the highest. These results verify the effectiveness of the proposed power-saving control scheme.

VI. EVALUATION OF LARGE CLOUD WITH 1,000 VMS

As explained in the previous section, the power-saving efficiency of the proposed scheme for a cloud system with 200 VMs was evaluated. In addition, as explained in this section, the times for determining VM reallocation and power-saving efficiency of a cloud system with 1,000 VMs for both the previously proposed system and the presently proposed system were evaluated.

A. Evaluation of time for determination of VM reallocation

1) Structure of evaluation system

The evaluation system includes 1,000 VMs and 400 servers in four DCs (as shown in Table VI). In addition, a WAN-management server, an inter-DC-management server, four DC/user VM-management servers are included. As for this system, switches, servers, and VMs in the DCs are emulated by open-source software, while switches in the WAN and management servers are real apparatuses.

TABLE VI. NUMBER OF PIECES OF ICT EQUIPMENT

	Item	Number of pieces of ICT equipment in 4 DCs
1	WAN-management server	1
2	SWs in WAN	3
3	DCs	4
4	Inter-DC-management server	1
5	DC/user VM-management servers	4
6	SWs in DCs	92
7	Servers in DCs	400
8	VMs on servers	1,000

2) Evaluation parameters and conditions

The main user VM-management server receives a list of multiple alternative servers from the sub user VM-management servers. The number of alternative servers is set as a parameter of the evaluation. As evaluation conditions, 768 VMs are migrated from original servers to other servers, and 232 VMs are not migrated. In addition, after the reallocation, the number of active servers is reduced from 400 to 84 (as listed in Table VII).

TABLE VII. NUMBER OF PIECES OF ICT EQUIPMENT AFTER VM REALLOCATION

	Number of DCs	Active SWs	Active servers	Number of VMs
1	DC 1	23	21	252
2	DC 2	23	21	252
3	DC 3	23	21	252
4	DC 4	23	21	244
	Total	92	84	1,000

3) Evaluation results

The evaluation results are shown in Figure 11. In the figure, the time for determining reallocation of 1,000 VMs is depicted. The horizontal axis shows the number of alternative servers per DC. Specifically, the number “1” means that only one alternative server is on the list sent from each other user VM-management server when the main user VM-management server requests an alternative server to migrate a VM to. On the other hand, the number “10” means that 10 alternative servers are on the list sent from each other user VM-management server. The vertical axis shows the time to determine reallocation of 1,000 VMs.

According to the results shown in Figure 11, in the case of our previous scheme with centralized management, over twenty-one minutes (1274 seconds) are needed to determine reallocation. On the other hand, in the case of the proposed distributed management scheme, the determination is done within about five minutes (250 seconds) when the main user VM-management server receives a list with one alternative server from each other user VM-management server. The determination time is about five times shorter than that taken by the previous system. Even if the main user VM-management server receives lists with five alternative servers, seven minutes (303 seconds) are enough to determine reallocation of 1,000 VMs.

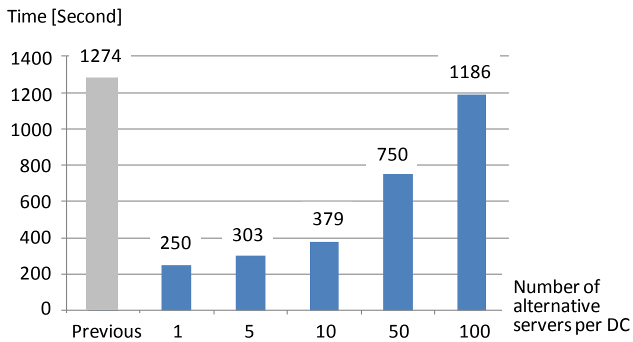


Figure 11. Time for determining VM reallocation

4) Discussion on performance of determining VM reallocation

In the previous centalized management scheme, the user VM-management server gathers information about future loads on all servers and SWs, and it determines one target server to migrate a VM to while guaranteeing QoS such as bandwidth to access a VM. On the other hand, in the proposed distributed management scheme, the user VM-management server sends requests for a list of alternative servers to other management servers in parallel. The load of the user VM-management server in the proposed system is therefore lower than that of the user VM-management server in the previous centalized-management-based system.

According to the evaluation results presented above, the proposed system has better performance in determining reallocation of 1,000 VMs. To move 1,000 VMs to other servers, it might take one hour (including the determination time for the reallocation). However, if 1,000 VMs are reallocated within one hour, it might be possible to reduce power consumption by about 30% because the proposed distibuted-control-based system easily changes the mode from normal operation to power-saving operation if the migration of 1,000 VMs is done within one hour. As shown by these evaluation results, the proposed system had enough time to reduce power consumption.

B. Evaluation of power saving for one week

The power-saving performance of the proposed system for one week was evaluated by assuming CPU and trafic loads based on a real-world server providing business applications.

1) Structure and conditions concerning evaluation system

The system used for evaluating the power-saving scheme is the same as that used in the evaluation mentioned above. However, as shown in Figures 12 to 15, models of CPU load and traffic are defined for one day in detail. In this evaluation, power-saving efficiencies for one day and one week were evaluated. Specifically, the power-saving efficiency on one weekday and one weekend day were evaluated. The power-saving efficiency for one week was also evaluated as total power saving over seven days. Specifically, total one-week power consumption was calculated by adding the power consumptions for five weekdays and one weekend.

2) Load modes and power-saving operation

Models of load and power consumption are shown in Figs. 12 and 13, which show a load model for a VM's CPU and a traffic model for VMs for one weekday, respectively. A load model for a VM's CPU and a traffic model for VMs for one weekend day are shown in Figs. 14 and 15, respectively. As shown in Figs. 12 and 13, high-load situations occur two times. Therefore, on the weekday, power-saving operation is done twice; however, on the weekend, the power-saving operation is done once.

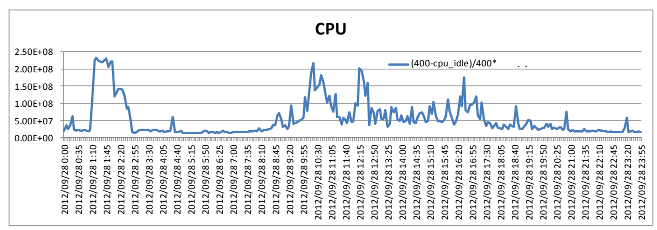


Figure 12. VM's CPU load model for a weekday

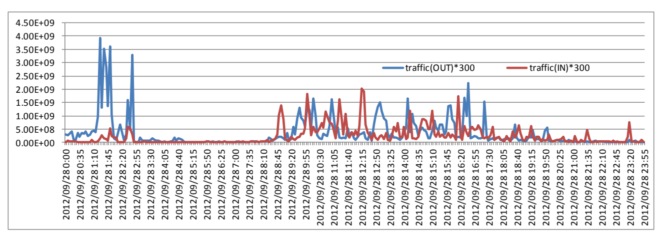


Figure 13. VMs' traffic models for a weekday

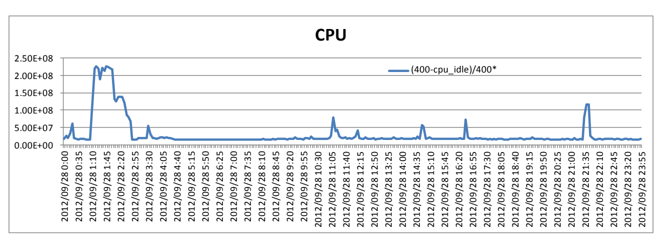


Figure 14. VM's CPU load model for a weekend day

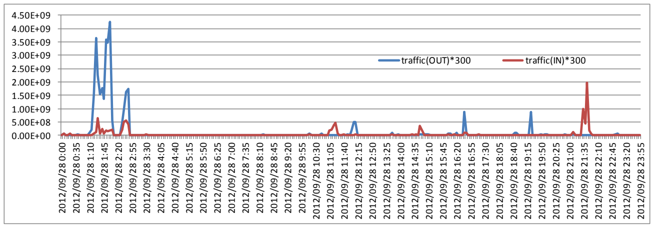


Figure 15. VMs' traffic models for a weekend day

3) Results of evaluation of power-saving control scheme

The effectiveness of the power-saving control scheme for one day and one week is shown in Table VIII. The reduction of energy consumption for one weekday is 24.3%, while the reduction for one weekend day is 46.6%. The energy consumption for one week is calculated by summing the energy consumptions for five weekdays and one weekend. In addition, the reduction for one week is 30.6%. As shown in Table VIII, the proposed system can reduce electric-energy consumption by 30%.

TABLE VIII. ELECTRIC-ENERGY CONSUMPTION OF A SYSTEM WITH 1,000 VMS

	Day	Without energy saving (kWh)	With energy saving (kWh)	Reduction (%)
1	1 weekday	2,076.707	1,572.646	24.3
2	1 weekend day	2,072.799	1,107.081	46.6
3	5 weekdays	10,383.535	7,863.230	24.3
4	1 weekend	4,145.598	2,214.162	46.6
5	1 week	14,529.133	10,077.392	30.6

C. Limitation of prototype system

The above-described evaluations verified a coordinated control function between networks' and servers' resources that maintains network QoS, such as network bandwidth consumed by a VM before it is migrated. In the evaluation, the QoS was controlled according to predicted future network traffic so that network congestion does not occur. From that perspective, a prototype system can guarantee the QoS. However, unpredicted network traffic occurs rarely. To guarantee the QoS of the network traffic with high priority even if unpredicted network traffic occurs, the system should create a transmission path with reserverd network bandwidth.

There is a tradeoff between perfect optimization of energy consumption and guaranteeing network QoS. To provide the perfect optimization, future traffic must be predicted perfectly. A level of the optimization depends on usage efficiency of network bandwidths. To increase the level of optimization, fewer network nodes are used. In that case, the risk of violation of the QoS increases. It therefore seems that there is an important balance between optimization of energy consumption and maintaining network QoS.

To evaluate energy saving and all communication overheads by management servers accurately, it is better to use a power meter for all network nodes and severs, since indirectly calculated power consumption was used in the present evaluation. In the proposed system, the effect of the management servers on communication overheads was not evaluated and remains as future work.

VII. CONCLUSION AND FUTURE WORK

A large-scale power-saving cloud system with a distributed-management scheme is proposed. The system is composed of multiple DCs connected by a WAN. It also includes an inter-DC-management server, multiple DC-management servers, and multiple user VM-management servers. In the proposed system, VMs are reallocated to reduce power consumption under condition of guaranteeing necessary CPU resources and network bandwidth for providing cloud services. Power saving covering the entire system is executed by cooperation between user VM-management servers and an inter-DC-management server.

The management scheme is based on distributed management instead of the central management of a previous system. In the proposed system, to distribute management load, each user VM-management server sends specifications of the intended VM to other user VM-management servers. The other user VM-management servers then independently return a list of alternative servers that can accommodate the intended VM. After receiving the lists, the user VM-management server selects the most suitable server to which the VM is migrated to and thus reduce electric-energy consumption of the entire system.

A prototype system, composed of 1,000 VMs, 400 servers, and four DCs, was developed and evaluated. The time for determining reallocation of 1,000 VMs is within five minutes, which is about five times shorter than that taken by the previous system. In addition, the evaluation results verify proper reallocation of VMs between DCs and the possibility of energy saving by approximately 30% for one-week cloud operation (under the conditions assumed in this evaluation).

For further study, the proposed power-saving cloud system will be evaluated by considering multiple real-world servers providing a diversity of business applications. In addition, it will be evaluated using power meters in consideration of the effect of multiple management servers on communication overheads. Besides, it will be implemented and evaluated by using real-world DC resources. As a result, it is expected that the proposed system will be enhanced from laboratory quality so that it can be applied as a real cloud system.

ACKNOWLEDGMENTS

Part of this research was supported by the MIC (The Japanese Ministry of Internal Affairs and Communications) projects "Research and Development on Signaling Technology of Network Configurations for Sustainable Environment" and "Research and Development on

Power-saving Communication Technology – Realization of the Eco-Internet".

REFERENCES

[1] T. Suzuki et al., "A Large-scale power-saving cloud system composed of multiple data centers," The Third International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY 2013), pp. 127-133, Mar. 2013.

[2] C. L. Belady, Microsoft Corporation, "Projecting annual new datacenter construction market size," Mar. 2011, http://cdn.globalfoundationservices.com/documents/Projecting_Annual_New_Data_Center_Construction_PDF.pdf [retrieved: May 2014].

[3] GreenTouch, "Our mission," http://www.greentouch.org/index.php?page=about-us/ [retrieved: May 2014].

[4] T. Suzuki et al., "Power-saving ICT platform that guarantees network bandwidth for cloud-service systems," TS-A4: Cloud Computing Technical Session, World Telecommunications Congress, Mar. 2012.

[5] VMware, Inc., "VMware distributed power management," http://www.vmware.com/resources/techresources/1080 [retrieved: May 2014].

[6] Xen Homepage, http://www.xen.org/ [retrieved: May 2014].

[7] M. Yamada, T. Yazaki, N. Matsuyama, and T. Hayashi, "Power efficient approach and performance control for routers," Proc. of IEEE International Conference on Communications (ICC Workshops 2009), pp. 1-5, June 2009.

[8] Y. Fukuda, T. Ikenaga, H. Tamura, M. Uchida, K. Kawahara, and Y. Oie, "Performance evaluation of power saving scheme with dynamic transmission capacity control," Proc. of IEEE Globecom Workshops 2009, pp. 1-5, Nov./Dec. 2009.

[9] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy consumption in optical IP networks," Journal of Lightwave Techology, vol. 27, no. 13, pp. 2391-2403, July 2009.

[10] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, "Energy consumption of telecommunication networks and related improvement options," IEEE Journal of selected topics in quantum electronics, vol. 17, no. 2, pp. 285-295, March/April 2011.

[11] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Energy efficiency in telecom optical networks," IEEE Communications surveys & tutorials, vol. 12, no. 4, pp. 441-458, Fourth quarter 2010.

[12] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy efficiency in the future internet: A servey of existing approaches and trends in energy-aware fixed nework infrastructures," IEEE Communications surveys & tutorials, vol. 13, no. 2, pp. 223-244, Second quarter 2011.

[13] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," Proc. of IEEE, vol. 99, no. 1, Jan. 2011.

[14] S. Yang, L. Chen, H. Tseng, H. Chung, and H. Lin, "Designing automatic power saving on virtualization environment," IEEE International Conference on Communication Technology (ICCT 2010), pp. 966-970, Nov. 2010.

[15] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," Proc. of IEEE Infocom 2012, pp. 1431-1439, Mar. 2012.

[16] T. Imada, M. Sato, and H. Kimura, "Power and QoS performance characteristics of virtualized servers," Proc. of IEEE/ACM International Conference on Grid Computing, pp. 232-240, Oct. 2009.

[17] B. Kantarci, L. Foschini, A. Corradi, and H.T. Mouftah, "Inter-and-intra data center VM-placement for energy-efficient large-Scale cloud systems," Proc. of IEEE Globecom Workshops on Management and Security technologies for Cloud Computing 2012, pp. 708-713, Dec. 2012

[18] A.Q. Lawey, T.E.H.El-Gorashi, and M.H. Elmirghani, "Distributed energy efficient clouds over core networks," Journal of Lightwave Techology, vol. 32, no. 7, pp. 1261-1281, April 2014.

[19] The Internet Engineering Task Force (IETF), "Energy management working group (EMAN WG) charter," http://datatracker.ietf.org/wg/eman/charter/ [retrieved: May 2014].

[20] Institute of Electrical and Electronics Engineers (IEEE), "IEEE802.1," http://www.ieee802.org/1/ [retrieved: May 2014].

[21] International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) Study Group 13 (SG13), "Question 21/13 – Future networks," http://www.itu.int/ITU-T/studygroups/com13/sg13-q21.html [retrieved: May 2014].

[22] Distributed Management Task Force, Inc. (DMTF), "Cloud management initiative," http://www.dmtf.org/standards/cloud [retrieved: May 2014].

[23] International Business Machines Corp. Homepage, http://www-01.ibm.com/software/analytics/spss/products/statistics/forecasting/ [retrieved: May 2014].

[24] T. Mukherjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, "Measurement-based power profiling of data center equipment", 2007 IEEE International Conference on Cluster Computing, pp.476-477, Sept. 2007.

[25] L.A. Barroso and U. Hölzle, "The case for energy-proportional computing," IEEE Computer, vol. 40, no. 12, pp.33-37, Dec. 2007.

Index Keys Method for Analyses of Urban Spaces

Methodological assumptions

Małgorzata Hanzl
Institute of Architecture and Town Planning
Lodz University of Technology
Lodz, Poland
e-mail: mhanzl@p.lodz.pl

***Abstract*— The paper proposes the index keys methodology of analysis of urban structures. The research stems from the descriptions of urban settings with regard to culture related issues. The geometrical analyses of public spaces, including the examination of streets and squares profiles and urban silhouettes, draw upon the writings of Kazimierz Wejchert, widely recognised for his contribution to the theory of urbanism in Poland. Cultural aspects of given settings require developing methods of description of public spaces. A few hypotheses are formulated concerning the relation between the crowd of people representing a given culture and the urban settings which constitute their *habitus*. Quantitative parameters describing the form of space are introduced, including: central angle, corrugation and regularity. The algorithmic method has been applied for the automatisation of the process. The preliminary results of analyses are presented as well as further research pathways.**

Keywords-urban design; public spaces; urban morphology.

I. INTRODUCTION

The requirement to establish the methodology of examination of physical urban settings has been recognised [1]. As it has been stated cities arise out of man's social needs, they are described as an embodiment of collective art and techniques [2]. Culture related aspects of everyday space usage are reflected first of all by internal organisation and arrangement of urban settings, their character and the meaning, which they convey. The issue of a theoretical exploration of the analysis of urban outdoor space should be addressed from a dual perspective: morphological and anthropological. The requirement to include culture related aspects of urban structures into the normative theory of urban design should be recognised and the epistemological apparatus included in the clearly defined ontology. The current paper attempts at the geometrical analyses of public spaces, including the examination of street and square profiles and urban silhouettes drawing upon the writings of Kazimierz Wejchert [3], a Polish post-war researcher and urban designer, widely recognised for his contribution to the theory of urbanism in Poland.

The paper is organised as follows: after this introduction, the anthropological perspective is briefly presented, which points out the understanding of situation and the definition of *habitus*, the main methodological assumptions are discussed, with an emphasis on the explanation of index keys concept. Section IV explores analytical methodology, introducing several parameters intended for the description of urban enclosures. Further on, the case study is introduced, which contains an explanation of the development of the methodology for detailed description of outdoor spaces and discusses the preliminary results of the research including the semi-automated analyses of selected parameters with Grasshopper. Section VI provides conclusions from the paper and presents further steps, which are to be taken in order to verify the presented methodology of analyses.

II. ANTHROPOLOGICAL PERSPECTIVE

When looking for the relation between urban structures and the culture of space usage there are three main issues that should be considered: (1) physical features, including distribution, shape and size of forms defining the space, (2) the distribution and behaviour of space users, which reflect their social order and (3) the flow of human movement, which finds its reflection in the sociometric layout of a given place. Flows are connected with movement/traffic and are related to space, following the definition by Yi Fu Tuan [4]. Concentrations enable contact and communication processes. They are static rather than dynamic, thus place related. Both types are closely interrelated, they inseparably interpenetrate each other. Whenever the human flow stops for a moment concentration occurs, though interrelations require more comfortable conditions to take place, among others: time and spatial arrangement. According to the theory formulated by Lynch [5], flows may be approached as paths and concentrations as nodes. Concentrations tend to a static form, while flows serve mainly as a means of getting to some destination. Taking into consideration mostly their static behaviour, the distribution of people in public spaces reflects social order. Contemporary research in anthropology clearly proves that lack of street or square boundary in modernists cities and districts is one of reasons for the absence of the traditional urban life. The form of physical enclosure should be subject of research to enable it to convey emotions and culture related meaning in a more conscious sense. The current paper is an extension of the former one [1], which elaborates more on the proposed methodology of examination of physical urban enclosures with the use of Grasshopper scripting for semi-automatised analyses.

A. *The concept of situation*

In anthropology, situation is defined as a theatre of human activities [6]. Goffman [7, p.18] refers to a situation as to *"the full spatial environment anywhere within which an entering person becomes a member of the gathering that is (or does then become) present"*. Anthropologists developed elaborated theories on ways in which a site is converted into a meaningful 'place', by inscribing human activities into the surroundings. The relationship between people and sites encompasses both: attaching meaning to space and *"recognition and cultural elaboration of perceived properties of environments in mutually constituting ways through narrative and praxis"* [8, p.14]. Schumacher [9] states that the role of architecture is to frame social communication [9, p.414]. Thomas, who introduced the concept of situation in the 1920s, defined it as a *"constellation of the factors determining the behaviour"* [10, p.8] after [9, p.420].

B. *The definition of habitus*

The morphological approach [11] refers the above concept to the urban structure introducing the notion of *habitus*. The set of identifiable cues, which may be qualified as culture-specific [12, pp.106-107], and referring to spaces, includes features like: *"quality, size, shape, enclosing elements, paving, barriers, and links, etc."*, requires examination with regard to the distribution of human flows and concentrations and their intensities, and consequently occasions for contacts. Both Gehl [13] and Whyte [14] point at similar rules of use of outside spaces. The territorial distribution and exchange of nonverbal cues serves the communication purpose and usually certain semantics may be attributed to it [7]. The behaviour of a given human group in concentrations reflects its culture. The movement component tends to be more universal and less culture dependent, as Hillier and Hanson [15] claim. The thesis is made that the rules, which govern the non-verbal communication component of the human group behaviour are the same ones that govern the distribution of buildings. They represent the same culture of space usage.

C. *Issues related to proxemics*

The proxemics approach, presented by Hall [16] and his successors, examines the relation between spatial patterns of space usage in different cultures and the material environment. The differences between morphological structures representing various cultures are particularly apparent in cities that, like Lodz, had become a melting pot of many cultures. Hall [17] identifies direct relationships between interpersonal distances and other characteristics of individuals and communities and the way they shape their own physical environment. Hillier and Hanson [15, p.27] refer to the usage of space and the patterns of behaviour appropriate for different communities and ethnic groups as the determinants of the final shape of urban structures. According to Hillier [18] a city is seen as a system of visual distances, strongly influenced by both perception and personal distances.

III. METHODOLOGICAL ASSUMPTIONS

A proposal of the method for the analysis of public spaces is presented, based on the writings of Wejchert [3]. K. Wejchert's theory of urban composition is widely recognised for its contribution to the theory of urbanism in Poland. Similarly to methods developed by Collins and his followers, it allows discussion of the atmosphere of urban spaces.

Forgoing morphological descriptions of urban structures were based on the analyses of plans, i.e., Conzenian school of urban morphology [19]. The other group refers to the diachronic characteristics of constructions, i.e., Muratori's tradition [20]. They do not allow for considerations referring to the ambiance of the settings discussed as a whole. The actual, practice-based approach engages the definition of genius loci, notably in rehabilitation projects.

The research applies the anthropological approach to the description of cities and urban structures. It follows and develops the so far available, descriptive methods. E.g., Rapoport listed a comprehensive set of culture related characteristics of physical structures [12, pp.106]. Hillier and Hanson [15, p.224] ponder on the method of investigating encounters as morphic languages. They conclude that the aim is to establish how encounter systems get differential properties. The resulting peculiarities would have different manifestations in space.

A. *Perception as a factor influencing the creation of space*

Strzemiński [21] pointed at the evolution of visual awareness along with the development of civilisation. Visual awareness was transformed together with the changes of socio-cultural settings. He examined it as a consequence of economic and technical development. Both above factors as well as the social structure of a community in the defined historical context influenced the way people perceive spatial settings. The notion of visual awareness is understood as the *"cooperation of seeing and thinking"*. It emphasises the role of cognitive absorption of perceived visual stimuli. Strzemiński [21] identifies two ways of the development of visual awareness. In rural cultures, it is the observation of the interior of an object, which finds its expression in the studies of nature. The second form was a silhouette vision. It is said to develop from the primitive contour observation in economies based on hunting and breeding animals. This form is typical of tribes accustomed to vast open spaces.

The derivative of the silhouette vision was the perspective of simple parallel projection. In the further stage it was followed by the development of rhythm, including architectural rhythmisation. The last phenomena was a consequence of the inclusion of the afterimage effect, natural for the perception of vast open spaces. Another form of seeing was the one concentrated on ware attributes, with emphasis on the texture and weight of objects. Usually this form was devoid of larger perspectives. It was particularly apparent in communities, whose main occupation was commerce. Adorno [22, p.5] points at the role of artworks as a medium reflecting the unconscious aspects of culture. He states that *"artworks are afterimages of empirical life insofar as they help the latter to what is denied them outside their own sphere and thereby free it from that to which they are* *condemned by reified external experience."* The same refers to the urban settings, which, perceived by a group of users, answer their needs, including the aesthetic criteria.

B. Rhythm - a component of unconscious contexting

Components containing the meaning of public spaces may 'speak' in different way. Some features are obvious and result from the functional conditions of a given development. They may be classified as direct communication. This group may include the elements of streets furniture such as: traffic lanes of various widths and surfaces, cycling paths, pavements, greenery, bollards, etc. It is a platitude to say that there are streets that are designed for driving fast and others inviting for a walk in the shade of trees. An elementary analysis of street profile allows one to distinguish a commercial street from a residential one. The general character of public spaces and their development through the ages is addressed in manuals of the history of urbanism.

Another part of communication may be classified as indirect, in analogy to non-verbal cues. According to Hall [23]: *"Nonverbal systems are closely tied to ethnicity (...) they are of the essence of ethnicity."* The consistency of urban pattern, as experienced in public spaces, is a consequence of the rules of crowd behaviour constituting part of a given culture. Kinesics is a way a person moves and handles their body. People specialised the language of the body making it integrated and congruent with everything they do, it is culturally determined and should be read against the given cultural background. Also, the presence of a synchronisation with settings is claimed, which takes place when the urbanscape belongs to the same culture as the visitor. Then a sense of belonging may be present and a place is perceived as more attractive than when the synchronisation is lacking. Settings which are out of phase are more likely to seem alien, unordered. As a consequence, the notion of rhythms may serve as an element connecting indirect communication with physical settings.

C. Theory of seeing – index keys concept

Like in the paintings of Van Gogh, seeing is concentrated around a few key points, which define how a scene is perceived [21]. The analyses should provide observation of processes: flows and forces, and concentrate on their key points. Situations that are the most important for the definition of cultural character, i.e., the moments of human interactions, particularly attract researchers' attention [17, p.56] – they are static rather than dynamic. The methodology of key points analogues to the anthropological method of taking photographs by members of the group that is observed, who are able to notice the clue activities important for their cultures and often unnoticeable for foreigners, allowing for observation of socially meaningful activities responsible for the formation of a cultural specific environment.

The application of the 'key points' methodology to the physical urban settings allows for the mathematical analyses of traditionally defined geometrical features of urban landscapes. It assumes the choice of the most obvious perspectives when observing the environment. In the case of urban spaces, it means choosing these view axes, which provide profiles and silhouettes respectively perpendicular or parallel to the main axe of a given path.

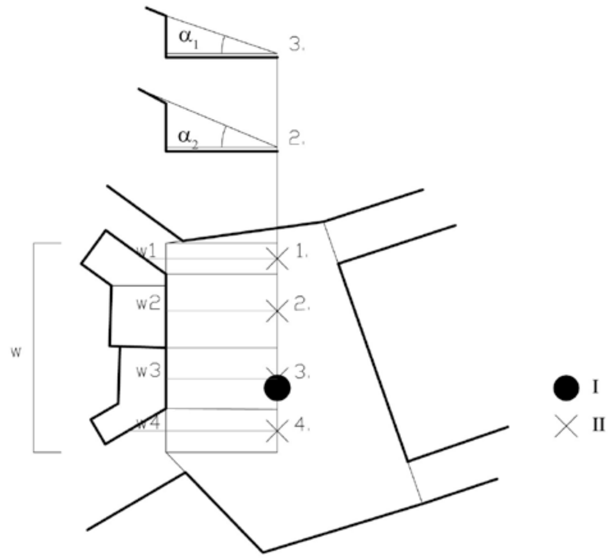


Figure 1. Index points and cross-sections definition. I – geometrical central point, II – index key points, α_n – central angles for each cross-sections, w_n – widths of each unique part of wall

In the proposed methodology the simplest way to extract cross-sections and silhouettes is used, based on orthogonal projection. The methods to acquire and analyse more realistic geometrical features including the perspective projection require further elaborations. The analyses of the profiles and of the silhouettes may use, among others, the highly efficient methodology proposed by Gal & Doytsher [24], which allows the replacement of the Line of Sight (LOS) methodology of extracting silhouettes of groups of buildings. The automatisation of profiles or silhouettes acquisition is also available using the Light of Sight (LOS) module of ArcScene by ESRI. There are also some preliminary attempts to model 3D isovist environment, in analogy to the method by Benedict [25] referring to the 2D plan drawing. These trials, some of them sophisticated, e.g., [26], equally require development of an analytical background.

Further methodology development should also take into consideration the processes described by the Gestalt psychology, e.g., the shortening of distances in the perception of distant buildings. Other field of studies are the ways in which humans perceive the environment, e.g., perception of meaningful entities [27].

IV. THE ANALYTICAL METHODOLOGY

A. Convex - definition

Hillier and Hanson [15] defined a series of rules governing the spatial order of analysed settlements. They noticed that the definition of the basic spatial unit for analyses, which would be distinguishable in the geometrical way, is essential for further considerations. Referring to their theoretical apparatus, a basic spatial unit that may serve for the description of public spaces is a convex. A *"fully convex fat space"* is defined as *"a part of a space, which represents the maximum extension of the point in the second dimension given the first dimension"* [15, p.91]. In Hillier logic of space, the implicit assumption is made that all the cells, representing spaces, are similar units, both in size and in shape. It does not describe the actual form of urban closures and the spatial edges are lacking. The critique concerns a lack of geometrical description of buildings, which form urban settings, including their size, shape and distribution (e.g., [28]).

Spaces, which are not defined spatially but by the presence of some other edges – like property borders, remain problematic. A more complete picture, which may serve to describe reality in a reliable way, requires the introduction of the shape and size parameter(s) and multiplying them by three dimensions. Studies in human perception show a trend to generalise objects to wholes, when the compounds are located close to each other, have similar attributes, may be described with the same contour line and their meaning, recognised from former experience, remains similar.

B. Description of the form of space

The way in which an observer perceives space in the urban interior depends on the parameters of cross-section. Wejchert [3]. The basic features important for describing convex spaces are profiles and wall silhouettes. The analysis of a wall's silhouette allows for identification of required index points, which may further on serve for the creation of profiles. Cross-sections may be created for any cue point of any unique physical form of objects surrounding the space change, i.e., the height and the shape of buildings.

Each index point is referred by one profile, various profiles require association with distinguished index points. The starting point for each profile is located on the line, which is parallel to the wall and goes through the geometrical centre of the given convex; see Fig. 1. Cross-sections are by definition perpendicular to the convex wall. In the case of buildings or other constructions that are set back from the convex edge and not perpendicular to it, the middle point of a building/construction is the location of an index point. Similar situation occurs in the case of buildings that are located behind other buildings but whose height exceeds the height of the front building.

The method may also serve for the description of some concavity closures. Yet, as their perception as one spatial unit is more the result of tradition than of their geometrical attributes, these shapes should be defined manually, i.e., divided into two or more basic convexes and then reconsidered as one whole. An example of a concavity space widely recognised as single urban interior is the L-shaped Piazza della Signoria in Florence.

C. Central angle

One of the most important parameters describing cross-sections is the central angle. The central angle is an angle between a horizontal plane parallel to the floor at the height of 1.5m (the medium level of sight for humans) and a line going through the highest point of the building defining the closure in a given index point. The point belongs both to the silhouette line and to the cross-section, see Fig. 2.

Wejchert [3] provides general rules for classification of closures based on the description of heritage sites, which are widely recognised as beautiful for their great proportions. The central angle values in most of the discussed squares range from 25° to 30°, e.g., Piazza Saint Marco in Venice - 28° to 30°, Old Market in Warsaw - 30°. The angle smaller than 10° refers to closures, which are feebly read in space. Either the plan dimensions are too vast or the vertical dimension is not adequate to provide the proper definition of space.

The closures of a central angle parameter higher than 60° rarely serve as public piazzas. An important feature for their evaluation are lighting conditions appropriate for a given climate. The general attitude towards more densely built spaces has changed recently, their values being widely recognised after the end of Modernism. The former pejorative connotation of terms such as "canyon" or "well" [3] lost their previous importance along with the common scarcity of defined spaces and dispersion of development. The central angle analysis is made for each of the cross-sections created at each of the index-points of the distinguished walls, and then combined for the walls forming the convexes, using the following formula (1), where α_1, α_2, α_3, α_n are values of central angles of each of the defined cross-sections, n is the number of index points for each wall, w_n is the width of a piece of a wall represented by a given index point and w is the length of the whole wall.

$$\alpha = \alpha_1 \times \frac{w_1}{w} + \alpha_2 \times \frac{w_2}{w} + \alpha_n \times \frac{w_n}{w} = \sum \left(\alpha_n \times \frac{w_n}{w} \right) \quad (1)$$

D. Corrugation and size

The urban spaces must be also measured using metric values. Humans, as Gehl asserts in the interview in a documentary film 'Urbanized' by Gary Hustwit, *"remain a small walking animal"* and require spaces of human scale. The spaces that are too large seem undefined. Gehl recognises a distance of 100m as a maximum that allows for proper reception by the observer of the environment. The assumed research methodology refers to the width of half of the closure, thus the distance should not exceed 50m. The actual dimensions of physical spaces reflect also the requirements defined by proxemics. The differences in personal distances influence both the perception of space and its production [16], [23], which means that we may assume that the size of space is perceived and designed differently by people of various cultural background. Continuing this thread, the analysis of the dimensions of public spaces proves that they remain culture specific.

The definition of space may be either precise or hazy. In the first case walls form clearly cut edges, in the second one buildings and other objects are scattered, forming a kind of fuzzy boundary. As Wejchert [3] argues, sight tends towards forms that are '*strong*', which means: clearly defined, and towards layouts that are concise. Parts or the whole of the observed constructions may be hidden behind other objects, which occurs both in the vertical as in the horizontal plane. In the case of breaks in the structure - i.e., openings in the walls, the closest object closing the perspective visible in the silhouette view is taken into consideration. Similarly, a higher building located in the background should be taken into account as, constituting a part of a silhouette, it influences the actual central angle parameter. The index points, where there are no visible constructions, are described with central angle value 0.

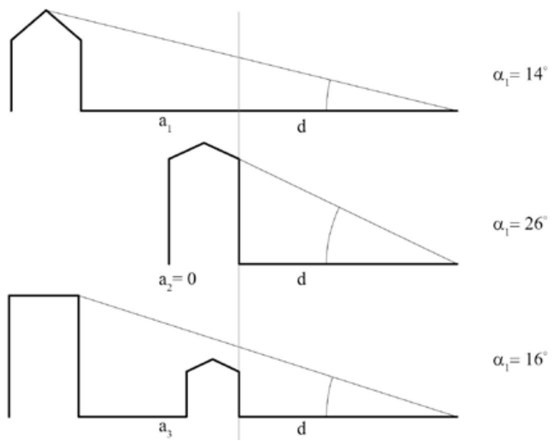


Figure 2. Corrugation of the wall, a_n- set back or behind of a part of a wall, d – a distance of the wall from the central point of the cross-section.

In a situation where buildings are set back from the line of frontages, the method allows for the description of an angle in a way similar to other cases. Variations of a buildings' offset are another parameter important for the definition of the space's character. The line of frontages may be located in the edge of a given convex or set back, the offset may be regular or irregular, any of these attributes influence the perception of the space (Fig. 2).

Corrugation may be defined using the formula (2), where φ symbolises corrugation value of the wall and γ - the offset of a single part of the wall. The possibilities of comparison of different situations are enabled thanks to the normalisation of offset values as in formula (3), where *a* represents the offset in metric units and *d* – the distance of the wall from the central point of the cross-section. In the case of some elements, offset of the lines of frontage shift should be given as positive numbers.

$$\varphi = \frac{\sum \gamma_n}{n} \tag{2}$$

$$\gamma_n = \frac{a_n}{d} \tag{3}$$

E. Distribution of index points

Further analyses include the distribution of index points, which reflects the distribution of buildings – each point belongs to a single building and the points are located in the middle of the facade. Such an analysis allows for easy detection of rhythms, repetitions, symmetries, axial layouts, etc. Distribution of index points may be described as clustered, spaced or scattered. It should be noticed that similar words are applied to the characteristics of the groups of people forming a crowd [29].

When analysing index point distribution the parameter of regularity may be defined referring to an ideal pattern, which for each case would mean equal distribution of the number of points defined for a given wall (Fig. 3). Any shift from the point resulting from an equal division should be measured and normalised by the width of the wall represented by each index point. The sum of all shifts divided by the number of index points describes the value of regularity for each wall. The regularity of the whole closure is described by the average value. The regularity may be described with the use of the formula (4), where τ is the regularity parameter, *r* represents a single shift, ϖ - width of an average part of a wall, *w* – width of a piece of a wall represented by a given index point and *n* is the number of index points for a given wall (5).

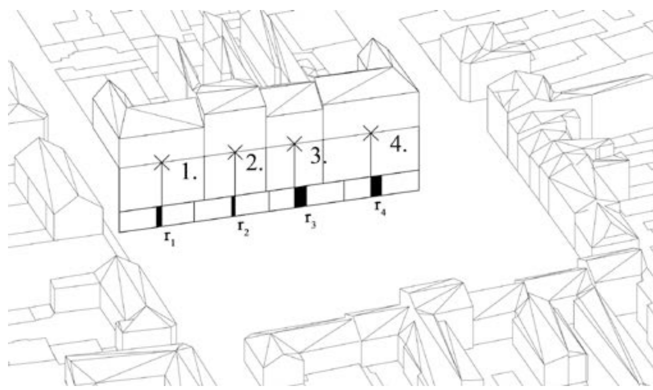


Figure 3. Regularity of the wall – method of description.

$$\tau = \frac{\sum r_n}{\varpi_n * n} \tag{4}$$

$$\varpi_n = \frac{\sum w_n}{n} \tag{5}$$

V. THE CASE STUDY

The case study pertains to two areas located in Lodz, Poland. One of them is Zgierska Street, located in the Old Town, in the former 'Jewish district'. In the 19th century, the district served as a habitat of the multiethnic society, in which orthodox Jews constituted a majority. These settings were commonly described as possessing a special ‘Jewish’ character. This notion is evoked by the form of public spaces, different than in other parts of the city. The other place is the Old Market in Lodz, located in the same neighbourhood. The subject of analysis is its former appearance, before the demolition during World War II and later. The other settings, used for reference, are an example of workers houses built for employees of the textile factory by Karol Scheibler in Plac Zwycięstwa as well as the first villa of this entrepreneur located on the other side of the same square. The square was later cut through by an important traffic route, Aleja Piłsudskiego. The remnants of the cultural heritage are however preserved even in the changed settings. Photographic analyses use the results of an inventory by students of the Institute of Architecture and Town Planning of Lodz University of Technology, 4th sem., tutor M. Hanzl.

The examples of the regularity analyses are presented in Fig. 4, results are included in Table I. The rhythm is described by the regularity parameter, which in case of Zgierska Street is lower than half and in the case of Plac Zwycięstwa is close to 0. This confirms the observation that the second case is a regular one in opposition to the first. The square was conceived as a single design, assuming repetition of identical workers buildings. The other site also remains regular although here it was not a requirement. We may assume that in this case a designer made a decision choosing this kind of solution appropriate in these settings.

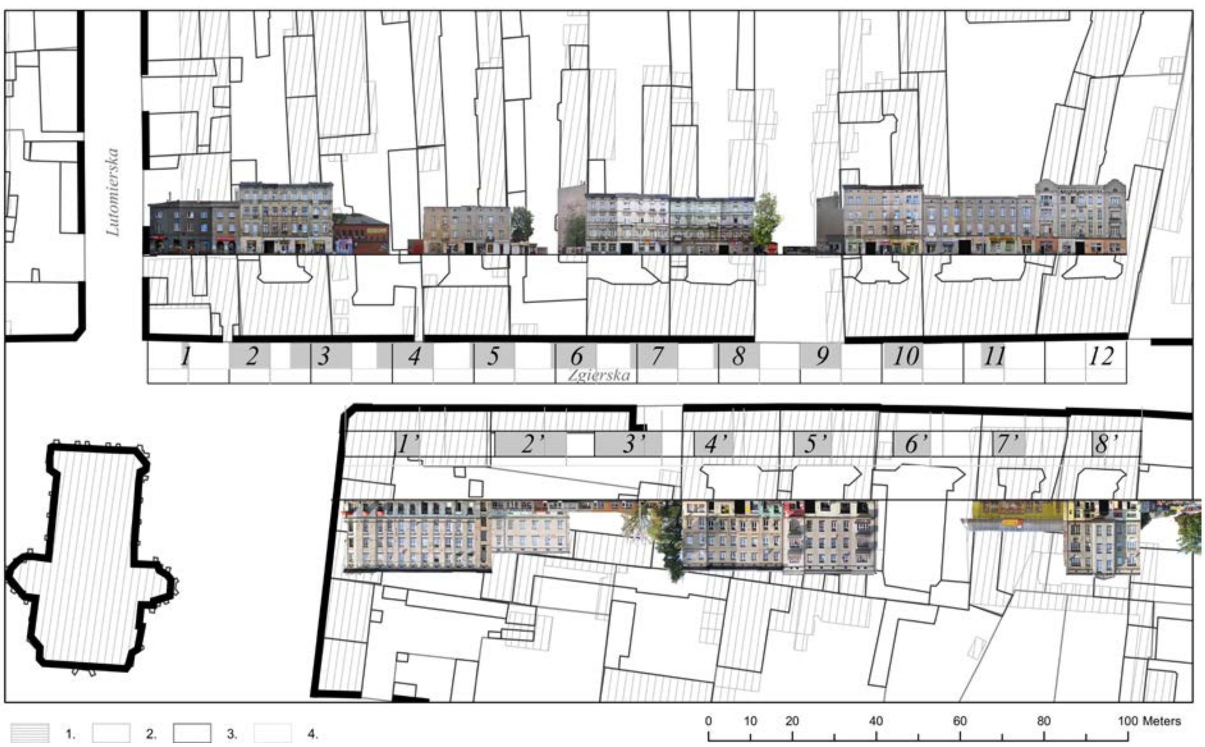


Figure 4. The analysis of regularity: current buildings (1) and parcels (2), historical buildings (3) and parcels (4). Values of shifts (r) for each key point shown in grey. Zgierska Street, Lodz, drawing uses results of inventory by students of the Institute of Architecture and Town Planning of Lodz University of Technology, 4th sem., tutor M. Hanzl

A. *Parametric approach*

In order to automatise the process of analysis, the Grasshopper for Rhinoceros 3D, a well known and widely recognised parametric modelling system, has been used. The analysis was performed on the sketchy reconstruction of the former appearance of the Old Town in Lodz, located in the very heart of the former so called Jewish quarter, not far from the first location analysed. The geometrical data has been acquired directly from the Sketchup model based on the archive photographs of the settings. The analyses of regularity and of central angle were performed. The value of corrugation has been counted as well. The details of the process of parametric modelling are presented in the illustrations (Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10). The numerical values are collected and ordered in the Table II and in the Table III. The summary values of central angle, corrugation and regularity are enbolden.

TABLE I. THE ANALYSIS OF REGULARITY: Τ - THE REGULARITY PARAMETER, R - A SINGLE SHIFT, W – WIDTH OF A SINGLE WALL, N - THE NUMBER OF INDEX POINTS FOR A GIVEN WALL

West side of Zgierska Street				East side of Zgierska Street			
n	*r [m]*	*w [m]*	*τ*	*N*	*r [m]*	*w [m]*	*τ*
1	1.59	16.37		1'	5.81	35.71	
2	9.80	6.73		2'	17.15	34.49	
3	14.72	22.03		3'	16.15	11.26	
4	13.55	19.48		4'	9.41	23.04	
5	10.77	20.47		5'	9.10	23.74	
6	10.93	19.31		6'	8.92	24.43	
7	11.48	19.71		7'	7.78	19.88	
8	11.05	20.26		8'	2.96	18.24	
9	9.68	21.59					
10	9.30	18.72					
11	5.56	26.64					
12	0.05	21.97					
r_n	9.04	19.44	0.46		9.66	23.85	0.41
σ	4.46	4.70			4.82	8.11	

TABLE II. THE ANALYSES' VALUES FOR THE RECONSTRUCTION OF THE OLD MARKET IN LODZ: CORRUGATION AND CNETRAL ANGLE

	North facade	West facade	East facade	South facade	
Corrugation (φ)	0.003623	0.101632	0.014923	0.163982	
	0.009263	0.22144	0.264982	0.289413	
	0.013601	0.13674	0.558645	0.380535	
	0.00796		0.830679	0.437859	
			0.841983	0.452478	
			0.506262	0.47934	
			1.078754	0.341997	
Average corrugation	0.0086118	0.153271	0.585175	0.363658	**0.27767889**
Central angle (α)	0.217827	0.134396	0.233878	0.236677	
	0.192346	0.083026	0.17665	0.197135	
	0.259558	0.143964	0.153952	0.157785	
	0.185947		0.258512	0.173245	
			0.163747	0.189091	
			0.143275	0.191368	
			0.141501	0.146875	
	21.391969	12.0462	18.16452	18.45967	**17.5155888**

TABLE III. THE ANALYSES' VALUES FOR THE RECONSTRUCTION OF THE OLD MARKET IN LODZ: REGULARITY

Regularity (τ)	1.274003	0.237062	0.028775	1.652843	
	0.891403	0.098132	0.893457	1.497418	
	3.035771	0.335194	2.577649	1.420905	
	2.653172		3.06436	4.331363	
			2.686072	8.04802	
			7.255301	6.223592	
			5.978173	1.241455	
	0.103669	0.007883	0.22197	0.339451	**0.16824325**

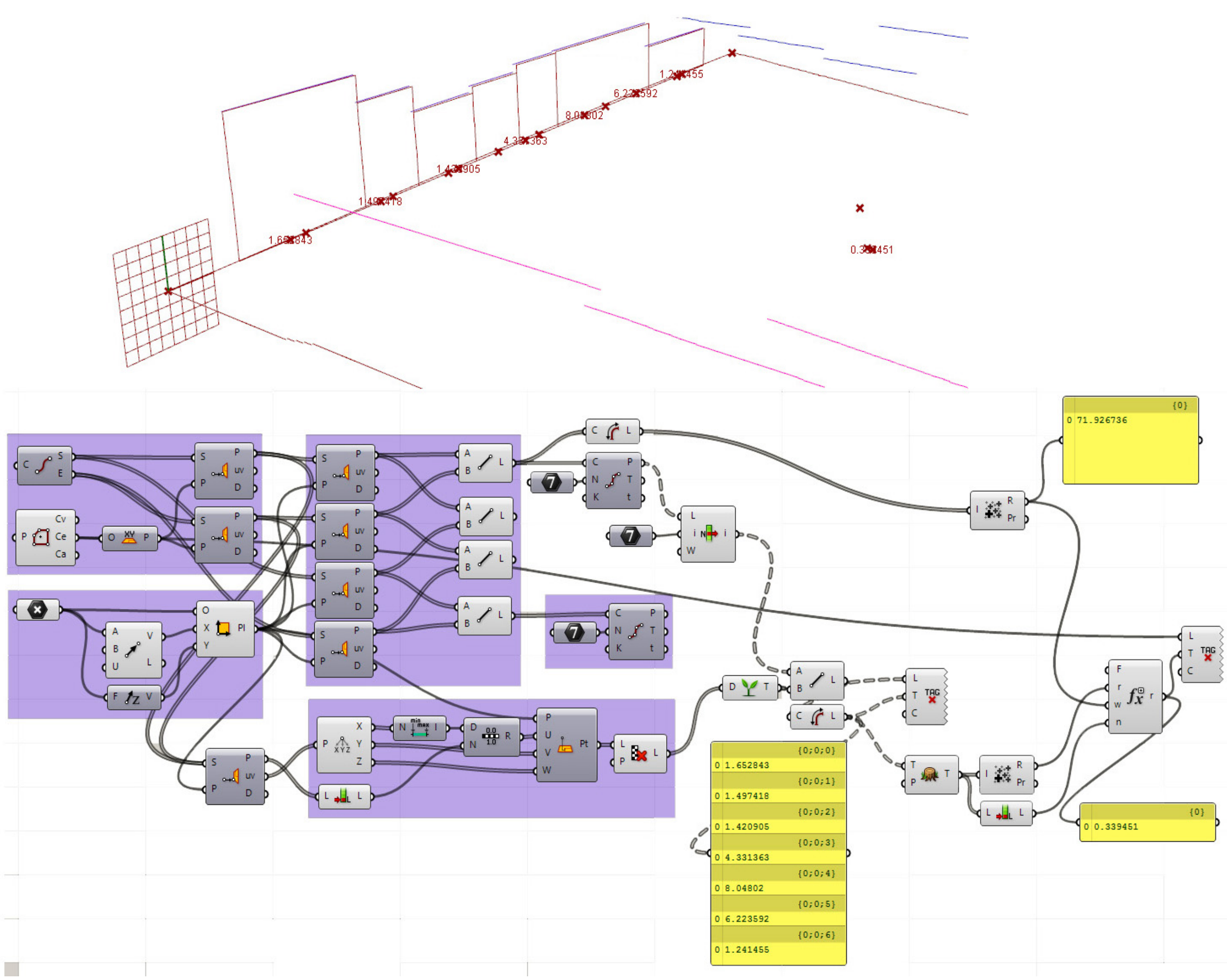


Figure 5. The analysis of regularity τ for the East facade of the Old Market in Lodz, reconstruction of the state before the World War II.

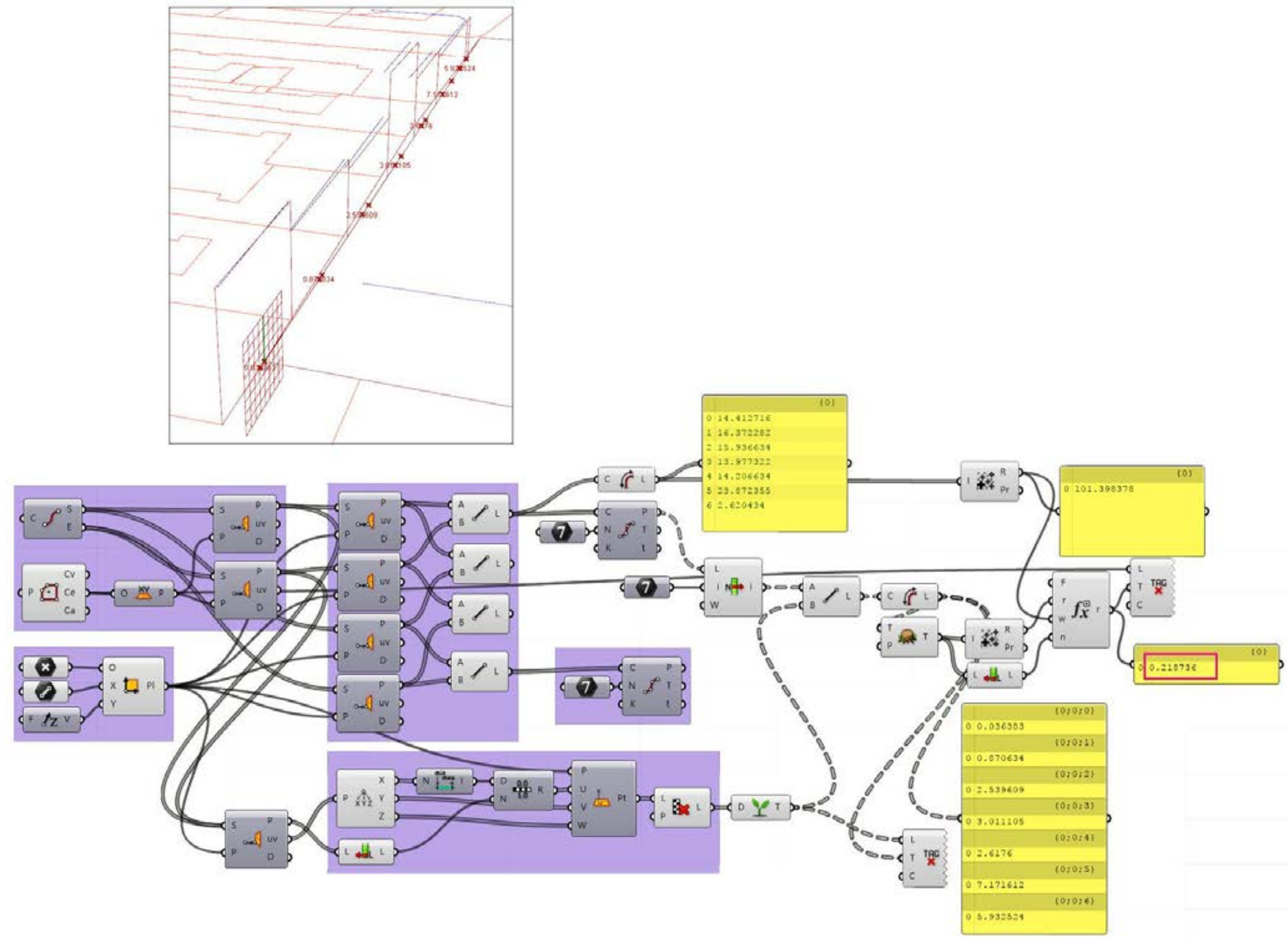


Figure 6. The analysis of regularity τ for the South facade of the Old Market in Lodz, reconstruction of the state before the World War II.

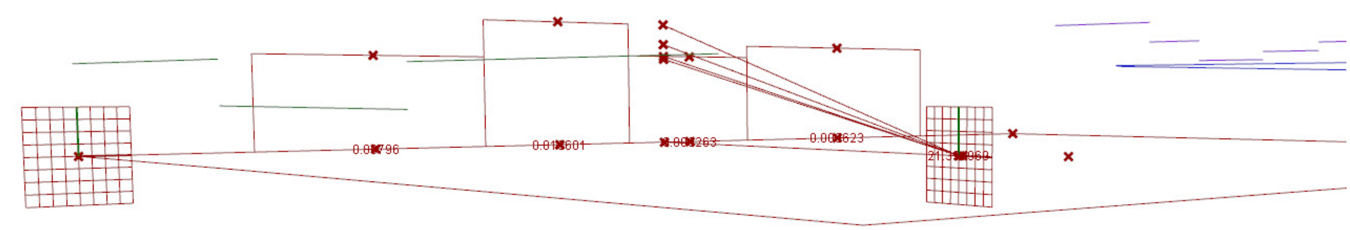


Figure 7. The analysis of the central angle α and corrugation φ values for the North facade of the Old Market in Lodz, reconstruction of the state before the World War II.

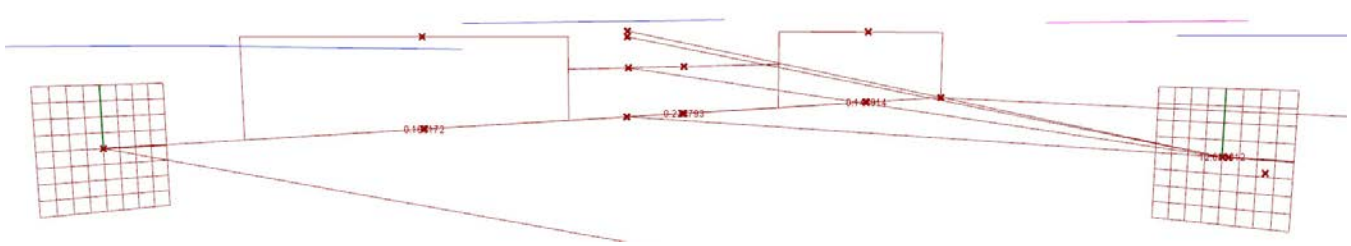


Figure 8. The analysis of the central angle α and corrugation φ values for the West facade of the Old Market in Lodz, reconstruction of the state before the World War II.

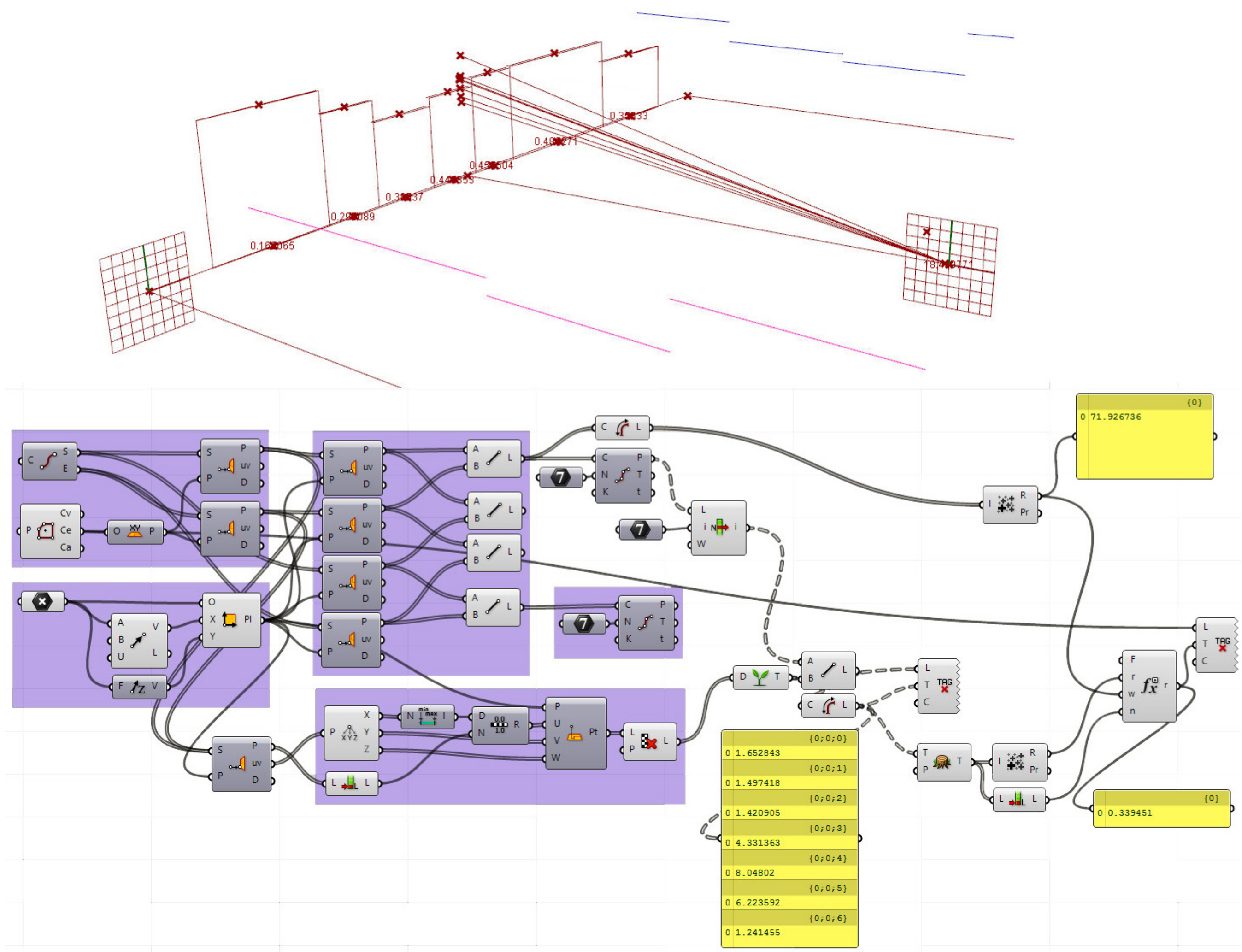


Figure 9. The analysis of the central angle and corrugation values for the East facade of the Old Market in Lodz, reconstruction of the state before the World War II.

The semi-automatic analyses presented in figures and tabular data collected as a result enabled work with large set of data and thus improvements of the method developed formerly and presented in [1]. Some small lapses were eliminated. Application of the semi-automatic method allowed significant acceleration of analyses. At this stage it is possible to switch between subsequent facades and analyses are performed automatically. The results will be further presented in the form of charts, showing rhythms of facades as waves, separate for each wall. Their overlap is going to be subject of the further analyses. There is also the possibility to write results of analyses to the separate Excel file for further research. For now this process is not fully automatic.

The analyses performed allow one to state that the corrugation of walls of the old market was significant, which, in comparison with the current settings, is an important difference. Similarly, the rhythms of current facades, constructed after World War II in socio-realism style, are very regular - the overall value is close to 0.

The application of an algorithmic method of analysis enables easy comparison of parameters for various settings. It has allowed verification of assumed mathematical apparatus and its validation. Use of Grasshopper, which is a the Rhino3D extension very popular among architects, may in future allow implementation of these types of analyses in the design process, similarly to other extensions of this type. The further work assumes creation of an extension for Grasshopper allowing the above analyses.

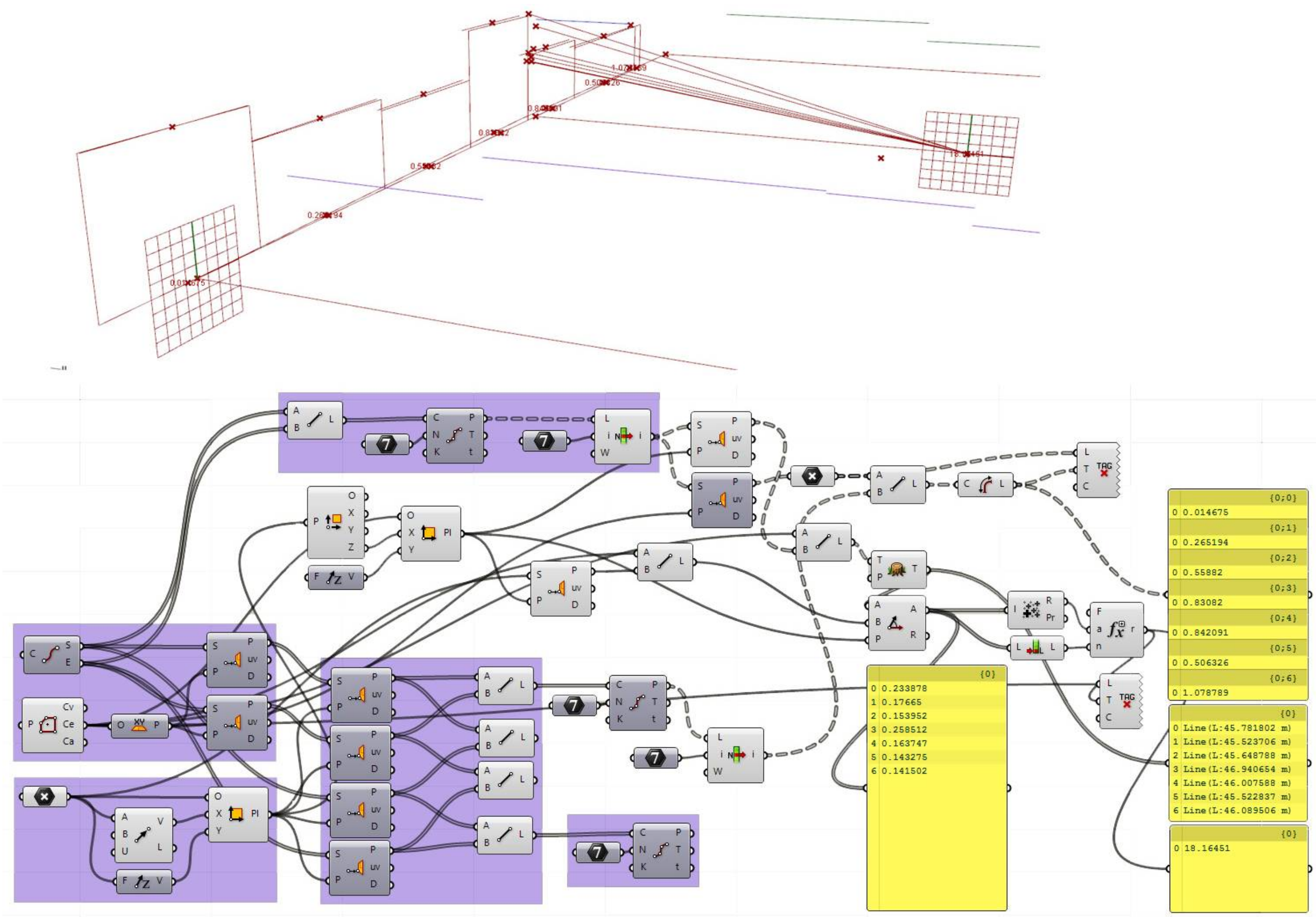


Figure 10. The analysis of the central angle α and corrugation values φ for the South facade of the Old Market in Lodz, reconstruction of the state before the World War II.

VI. Conclusions and Future Research

The challenge defined by Hall [23, p.55] that in a globalising world man must find out how *"basic cultural systems such as time and space are used to organise behaviour"* starts to influence contemporary urban design thought, as numerous studies show [9]. The thread of cultural studies imports viable content to the proposal of ontology for urban design, which is being developed, e.g., by Duarte et al. [30], Beirão et al. [31] or Beirão [32]. The requirement to define the methodology of description of public space character has been recognised. The studies of urban morphology are going through a period of intensive revival after a break associated with the activities of modernists [33] and attract the attention of numerous researchers all over the world, as Gauthier and Gilliland [9] describe in their comprehensive résumé.

An extensive set of culture dependent features was defined by Rapoport [12]. The current study provides assumptions to the quantitative description of public spaces based on the theory by Wejchert [3]. The concept of index points is introduced, which enables examination of the physical form of urban settings with the use of geometrical description. Basic values are defined, including the parameters of central angle, regularity and corrugation of an enclosure. Further development of the current theory is envisaged, including different approaches to the analyses of urban silhouettes and cross-sections, as well as its verification for the description of the assumed case study.

The current research is an ongoing one. Further steps include validation of the proposed methodology in an experimental way and comparison of various urban environments. This may help to understand the diachronic aspects of urban development. The first step covered exploratory modeling of various urban environments with the use of available software, including: (1) Google SketchUp; (2) CityEngine; (3) Rhino and comparison of results with the two-dimensional analysis explained hitherto [19]. The further steps assume comparison of resulting values with the extended analyses of kinetics patterns.

Acknowledgment

The current study is a part of the research project financed by National Science Centre of Poland UMO-2011/03/D/HS3/01630, entitled: "*Morphological analysis of urban structures – the cultural approach. Case studies of Jewish communities in the chosen settlements of Lodz and Masovian voivodeships*".

REFERENCES

[1] M. Hanzl, "Index keys method for analyses of urban public spaces," The Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2013) IARIA, Feb. 2013, pp. 108-113, ISSN: 2308-393X, ISBN: 978-1-61208-251-6.

[2] L. Mumford, The Culture of Cities. New York: Harcourt Brace, Jovanovich, Inc., 1970.

[3] K. Wejchert, Elementy kompozycji urbanistycznej. Warszawa: Wydawnictwo Arkady, 1984.

[4] Y.-F. Tuan, Space and Place, The Perspective of Experience. Minneapolis, London: University of Minnesota Press, 1977.

[5] K. Lynch, The Image of the City. Cambridge: The MIT Press, 1960.

[6] R. S. Perinbanayagam, “The definition of the situation: an analysis of the ethnomethodological and dramaturgical view,” The Sociological Quarterly, vol. 15, pp. 521–541, 1974.

[7] E. Goffman, The Presentation of Self in Everyday Life, Monograph. Edinburgh: University of Edinburgh, Social Science Research Centre, 1959.

[8] D. L. Lawrence and S. M. Low, “Locating Culture,” The antropology of Space and Place, Locating Culture, S. M. Low and D. L. Lawrence-Zuniga, Eds. Oxford: Blackwell Publishing, 2009, pp. 1–47.

[9] P. Schumacher, The Autopoiesis of Architecture A New Framework for Architecture, vol. 1. Great Britain: A John Wiley and Sons, Ltd, Publication, 2011.

[10] W. I. Thomas, Primitive Behavior: An Introducton to the Social Sciences. New York: McGraw-Hill, 1937.

[11] P. Panerai, J. Ch. Depaule, and M. Demorgon, Analyse urbaine, Marseille: Édition Parenthèses, 2009.

[12] A. Rapoport, The Meaning of the Built Environment. A Nonverbal Communication Approach. Tuscon: The University of Arizona Press, 1990.

[13] J. Gehl, Life Between Buildings, Using Public Space. Kopenhagen: The Danish Architectural Press, 2001.

[14] W. Whyte, The Social Life of Small Urban Places. New York: Project for Public Spaces, 2009.

[15] B. Hillier and J. Hanson, The Social Logic of Space, Cambridge: Cambridge University Press, 1984

[16] E. T. Hall, Hidden Dimension. Garden City, NY: Doubleday, 1966.

[17] E. T. Hall, “Proxemics,” The Anthropology of Space and Place, Locating Culture, S. M. Low and D. L. Lawrence-Zuniga, Eds. Oxford: Blackwell Publishing, pp. 51–73, 2009.

[18] B. Hillier, “The genetic code for cities – is it simpler than we thought?,” Complexity Theories of Cities have come of Age, Delft: TU Delft, Sep. 2009.

[19] J. W. R. Whitehead and P. J. Larkham, Eds., Urban Landscapes, International Perspectives. London: Routledge, 2000.

[20] G. Cataldi, G. L. Maffei, and P. Vaccaro, “Saverio Muratori and the Italian school of planning typology,” Urban Morphology, vol. 6, no. 1, pp. 3–12, 2002.

[21] W. Strzemiński, Teoria Widzenia. Kraków: Wydawnictwo Literackie, 1974.

[22] T. W. Adorno, Aesthetic Theory. London, New York: Continuum International Publishing Group, 2011.

[23] E. T. Hall, Beyond Culture. New York: Anchor Books, 1989.

[24] O. Gal and Y. Doytsher, “Fast and accurate visibility computation in a 3D urban environment,” in The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2012), IARIA, Jan. 2012, pp. 105–110, ISSN: 2308-393X, ISBN: 978-1-61208-178-6.

[25] M. Benedikt, "To take hold of space: isovists and isovists fields," Environment and Planning B: Planning and Design, vol. 6, pp. 647–65, 1979.

[26] E. Morello and C. Ratti, "A digital image of the city: 3D isovists in Lynch’s urban analysis." Environment and Planning B: Planning and Design 36:5, pp.837-853, 2009.

[27] S. Guberman, V. V. Maximov, and A. Pashintsev, “Gestalt and image understanding,” Gestalt Theory, vol. 34, no. 2, pp. 143–166, 2012.

[28] C. Ratti, “Space syntax: some inconsistencies,” Environment and Planning B: Planning and Design, vol. 31, no. 4, pp. 487–499, 2004.

[29] N. Fridman and G. A. Kaminka, “Towards a cognitive model of crowd behavior based on social comparison theory,” Proceedings of the national conference on artificial intelligence, Vol. 22. No. 1. Menlo Park, CA; Cambridge, MA; London: AAAI Press, MIT Press, p.731, 2007.

[30] J. P. Duarte, N. Montenegro, and J. N. Beirão, “Public space patterns Towards a CIM standard for urban public space,” Respecting Fragile Places, Proceedings of the 29th Conference on Education in Computer Aided Architectural Design in Europe, T. Zupancic, M. Juvancic, S. Verovsek, & A. Jutraz, Eds. Ljubljana, Slovenia: eCAADe, pp. 79–85, 2011.

[31] J.N. Beirão, J.P. Duarte, N. Montenegro, "Public space patterns towards a CIM standard for urban public space," Respecting Fragile Places, Proceedings of the 29th Conference on Education in Computer Aided Architectural Design in Europe, T. Zupancic, M. Juvancic, S. Verovsek, & A. Jutraz, Eds. Ljubljana, Slovenia: eCAADe, pp. 79–85, 2011.

[32] J.N. Beirão, CItyMaker Designing Grammars for Urban Design, Architecture and the Built Environment, Vol.5, Delft University of Technology, 2012.

[33] J. Rykwert, The Idea of a Town, The Anthropology of Urban Form in Rome, Italy and the Ancient World. Cambridge, London: The MIT Press, 1989.

[34] P. Gauthier and J. Gilliland, ‘‘Mapping urban morphology : a classification scheme for interpreting contributions to the study of urban form,” Urban Morphology, vol. 10, no. 1, pp. 41–50, 2006.

www.iariajournals.org

International Journal On Advances in Intelligent Systems

ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, ENERGY, COLLA, IMMM, INTELLI, SMART, DATA ANALYTICS

issn: 1942-2679

International Journal On Advances in Internet Technology

ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING, MOBILITY, WEB

issn: 1942-2652

International Journal On Advances in Life Sciences

eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO, SOTICS, GLOBAL HEALTH

issn: 1942-2660

International Journal On Advances in Networks and Services

ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION, VEHICULAR, INNOV

issn: 1942-2644

International Journal On Advances in Security

ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

issn: 1942-2636

International Journal On Advances in Software

ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, IMMM, MOBILITY, VEHICULAR, DATA ANALYTICS

issn: 1942-2628

International Journal On Advances in Systems and Measurements

ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL, INFOCOMP

issn: 1942-261x

International Journal On Advances in Telecommunications

AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA, COCORA, PESARO, INNOV

issn: 1942-2601