

International Journal on

Advances in Software



2015 vol. 8 nr. 3&4

The *International Journal on Advances in Software* is published by IARIA.

ISSN: 1942-2628

journals site: <http://www.ariajournals.org>

contact: petre@aria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Software, issn 1942-2628
vol. 8, no. 3 & 4, year 2015, <http://www.ariajournals.org/software/>

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Software, issn 1942-2628
vol. 8, no. 3 & 4, year 2015,<start page>:<end page> , <http://www.ariajournals.org/software/>

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA

www.aria.org

Copyright © 2015 IARIA

Editor-in-Chief

Luigi Lavazza, Università dell'Insubria - Varese, Italy

Editorial Advisory Board

Hermann Kaindl, TU-Wien, Austria

Herwig Mannaert, University of Antwerp, Belgium

Editorial Board

Witold Abramowicz, The Poznan University of Economics, Poland

Abdelkader Adla, University of Oran, Algeria

Syed Nadeem Ahsan, Technical University Graz, Austria / Iqra University, Pakistan

Marc Aiguier, École Centrale Paris, France

Rajendra Akerkar, Western Norway Research Institute, Norway

Zaher Al Aghbari, University of Sharjah, UAE

Riccardo Albertoni, Istituto per la Matematica Applicata e Tecnologie Informatiche "Enrico Magenes" Consiglio Nazionale delle Ricerche, (IMATI-CNR), Italy / Universidad Politécnica de Madrid, Spain

Ahmed Al-Moayed, Hochschule Furtwangen University, Germany

Giner Alor Hernández, Instituto Tecnológico de Orizaba, México

Zakarya Alzamil, King Saud University, Saudi Arabia

Frederic Amblard, IRIT - Université Toulouse 1, France

Vincenzo Ambriola, Università di Pisa, Italy

Andreas S. Andreou, Cyprus University of Technology - Limassol, Cyprus

Annalisa Appice, Università degli Studi di Bari Aldo Moro, Italy

Philip Azariadis, University of the Aegean, Greece

Thierry Badard, Université Laval, Canada

Muneera Bano, International Islamic University - Islamabad, Pakistan

Fabian Barbato, Technology University ORT, Montevideo, Uruguay

Peter Baumann, Jacobs University Bremen / Rasdaman GmbH Bremen, Germany

Gabriele Bavota, University of Salerno, Italy

Grigorios N. Beligiannis, University of Western Greece, Greece

Noureddine Belkhatir, University of Grenoble, France

Jorge Bernardino, ISEC - Institute Polytechnic of Coimbra, Portugal

Rudolf Berrendorf, Bonn-Rhein-Sieg University of Applied Sciences - Sankt Augustin, Germany

Ateet Bhalla, Independent Consultant, India

Fernando Boronat Seguí, Universidad Politecnica de Valencia, Spain

Pierre Borne, Ecole Centrale de Lille, France

Farid Bourennani, University of Ontario Institute of Technology (UOIT), Canada

Narhimene Boustia, Saad Dahlab University - Blida, Algeria

Hongyu Pei Breivold, ABB Corporate Research, Sweden
Carsten Brockmann, Universität Potsdam, Germany
Antonio Bucchiarone, Fondazione Bruno Kessler, Italy
Georg Buchgeher, Software Competence Center Hagenberg GmbH, Austria
Dumitru Burdescu, University of Craiova, Romania
Martine Cadot, University of Nancy / LORIA, France
Isabel Candal-Vicente, Universidad del Este, Puerto Rico
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Jose Carlos Metrolho, Polytechnic Institute of Castelo Branco, Portugal
Alain Casali, Aix-Marseille University, France
Yaser Chaaban, Leibniz University of Hanover, Germany
Patrik Chamuczyński, Radytek, Poland
Savvas A. Chatzichristofis, Democritus University of Thrace, Greece
Antonin Chazalet, Orange, France
Jiann-Liang Chen, National Dong Hwa University, China
Shiping Chen, CSIRO ICT Centre, Australia
Wen-Shiung Chen, National Chi Nan University, Taiwan
Zhe Chen, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
PR
Po-Hsun Cheng, National Kaohsiung Normal University, Taiwan
Yoonsik Cheon, The University of Texas at El Paso, USA
Lau Cheuk Lung, INE/UFSC, Brazil
Robert Chew, Lien Centre for Social Innovation, Singapore
Andrew Connor, Auckland University of Technology, New Zealand
Rebeca Cortázar, University of Deusto, Spain
Noël Crespi, Institut Telecom, Telecom SudParis, France
Carlos E. Cuesta, Rey Juan Carlos University, Spain
Duilio Curcio, University of Calabria, Italy
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Paulo Asterio de Castro Guerra, Tapijara Programação de Sistemas Ltda. - Lambari, Brazil
Cláudio de Souza Baptista, University of Campina Grande, Brazil
Maria del Pilar Angeles, Universidad Nacional Autónoma de México, México
Rafael del Vado Vírseada, Universidad Complutense de Madrid, Spain
Giovanni Denaro, University of Milano-Bicocca, Italy
Nirmit Desai, IBM Research, India
Vincenzo Deufemia, Università di Salerno, Italy
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Javier Diaz, Rutgers University, USA
Nicholas John Dingle, University of Manchester, UK
Roland Dodd, CQUniversity, Australia
Aijuan Dong, Hood College, USA
Suzana Dragicevic, Simon Fraser University- Burnaby, Canada
Cédric du Mouza, CNAM, France
Ann Dunkin, Palo Alto Unified School District, USA
Jana Dvorakova, Comenius University, Slovakia
Lars Ebrecht, German Aerospace Center (DLR), Germany

Hans-Dieter Ehrich, Technische Universität Braunschweig, Germany
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Atilla Elçi, Aksaray University, Turkey
Khaled El-Fakih, American University of Sharjah, UAE
Gledson Elias, Federal University of Paraíba, Brazil
Sameh Elnikety, Microsoft Research, USA
Fausto Fasano, University of Molise, Italy
Michael Felderer, University of Innsbruck, Austria
João M. Fernandes, Universidade de Minho, Portugal
Luis Fernandez-Sanz, University of de Alcalá, Spain
Felipe Ferraz, C.E.S.A.R, Brazil
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Wolfgang Fohl, Hamburg University, Germany
Simon Fong, University of Macau, Macau SAR
Gianluca Franchino, Scuola Superiore Sant'Anna, Pisa, Italy
Naoki Fukuta, Shizuoka University, Japan
Martin Gaedke, Chemnitz University of Technology, Germany
Félix J. García Clemente, University of Murcia, Spain
José García-Fanjul, University of Oviedo, Spain
Felipe Garcia-Sanchez, Universidad Politecnica de Cartagena (UPCT), Spain
Michael Gebhart, Gebhart Quality Analysis (QA) 82, Germany
Tejas R. Gandhi, Virtua Health-Marlton, USA
Andrea Giachetti, Università degli Studi di Verona, Italy
Robert L. Glass, Griffith University, Australia
Afzal Godil, National Institute of Standards and Technology, USA
Luis Gomes, Universidade Nova Lisboa, Portugal
Diego Gonzalez Aguilera, University of Salamanca - Avila, Spain
Pascual Gonzalez, University of Castilla-La Mancha, Spain
Björn Gottfried, University of Bremen, Germany
Victor Govindaswamy, Texas A&M University, USA
Gregor Grambow, University of Ulm, Germany
Carlos Granell, European Commission / Joint Research Centre, Italy
Christoph Grimm, University of Kaiserslautern, Austria
Michael Grottko, University of Erlangen-Nuernberg, Germany
Vic Grout, Glyndwr University, UK
Ensar Gul, Marmara University, Turkey
Richard Gunstone, Bournemouth University, UK
Zhensheng Guo, Siemens AG, Germany
Ismail Hababeh, German Jordanian University, Jordan
Shahliza Abd Halim, Lecturer in Universiti Teknologi Malaysia, Malaysia
Herman Hartmann, University of Groningen, The Netherlands
Jameleddine Hassine, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Peizhao Hu, NICTA, Australia
Chih-Cheng Hung, Southern Polytechnic State University, USA
Edward Hung, Hong Kong Polytechnic University, Hong Kong

Noraini Ibrahim, Universiti Teknologi Malaysia, Malaysia
Anca Daniela Ionita, University "POLITEHNICA" of Bucharest, Romania
Chris Ireland, Open University, UK
Kyoko Iwasawa, Takushoku University - Tokyo, Japan
Mehrshid Javanbakht, Azad University - Tehran, Iran
Wassim Jaziri, ISIM Sfax, Tunisia
Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Jinyuan Jia, Tongji University. Shanghai, China
Maria Joao Ferreira, Universidade Portucalense, Portugal
Ahmed Kamel, Concordia College, Moorhead, Minnesota, USA
Teemu Kanstrén, VTT Technical Research Centre of Finland, Finland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Ayad ali Keshlaf, Newcastle University, UK
Nhien An Le Khac, University College Dublin, Ireland
Sadegh Kharazmi, RMIT University - Melbourne, Australia
Kyoung-Sook Kim, National Institute of Information and Communications Technology, Japan
Youngjae Kim, Oak Ridge National Laboratory, USA
Roger "Buzz" King, University of Colorado at Boulder, USA
Cornel Klein, Siemens AG, Germany
Alexander Knapp, University of Augsburg, Germany
Radek Koci, Brno University of Technology, Czech Republic
Christian Kop, University of Klagenfurt, Austria
Michal Krátký, VŠB - Technical University of Ostrava, Czech Republic
Narayanan Kulathuramaiyer, Universiti Malaysia Sarawak, Malaysia
Satoshi Kurihara, Osaka University, Japan
Eugenijus Kurilovas, Vilnius University, Lithuania
Philippe Lahire, Université de Nice Sophia-Antipolis, France
Alla Lake, Linfo Systems, LLC, USA
Fritz Laux, Reutlingen University, Germany
Luigi Lavazza, Università dell'Insubria, Italy
Fábio Luiz Leite Júnior, Universidade Estadual da Paraíba, Brazil
Alain Lelu, University of Franche-Comté / LORIA, France
Cynthia Y. Lester, Georgia Perimeter College, USA
Clement Leung, Hong Kong Baptist University, Hong Kong
Weidong Li, University of Connecticut, USA
Corrado Loglisci, University of Bari, Italy
Francesco Longo, University of Calabria, Italy
Sérgio F. Lopes, University of Minho, Portugal
Pericles Loucopoulos, Loughborough University, UK
Alen Lovrencic, University of Zagreb, Croatia
Qifeng Lu, MacroSys, LLC, USA
Xun Luo, Qualcomm Inc., USA
Shuai Ma, Beihang University, China
Stephane Maag, Telecom SudParis, France
Ricardo J. Machado, University of Minho, Portugal
Maryam Tayefeh Mahmoudi, Research Institute for ICT, Iran

Nicos Malevris, Athens University of Economics and Business, Greece
Herwig Mannaert, University of Antwerp, Belgium
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Francesco Marcelloni, University of Pisa, Italy
Eda Marchetti, Consiglio Nazionale delle Ricerche (CNR), Italy
Gerasimos Marketos, University of Piraeus, Greece
Abel Marrero, Bombardier Transportation, Germany
Adriana Martin, Universidad Nacional de la Patagonia Austral / Universidad Nacional del Comahue, Argentina
Goran Martinovic, J.J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
Stephan Mäs, Technical University of Dresden, Germany
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Jose Merseguer, Universidad de Zaragoza, Spain
Seyedeh Leili Mirtaheri, Iran University of Science & Technology, Iran
Lars Moench, University of Hagen, Germany
Yasuhiko Morimoto, Hiroshima University, Japan
Antonio Navarro Martín, Universidad Complutense de Madrid, Spain
Filippo Neri, University of Naples, Italy
Muaz A. Niazi, Bahria University, Islamabad, Pakistan
Natalja Nikitina, KTH Royal Institute of Technology, Sweden
Roy Oberhauser, Aalen University, Germany
Pablo Oliveira Antonino, Fraunhofer IESE, Germany
Rocco Oliveto, University of Molise, Italy
Sascha Opletal, Universität Stuttgart, Germany
Flavio Oquendo, European University of Brittany/IRISA-UBS, France
Claus Pahl, Dublin City University, Ireland
Marcos Palacios, University of Oviedo, Spain
Constantin Paleologu, University Politehnica of Bucharest, Romania
Kai Pan, UNC Charlotte, USA
Yiannis Papadopoulos, University of Hull, UK
Andreas Papasalouros, University of the Aegean, Greece
Rodrigo Paredes, Universidad de Talca, Chile
Päivi Parviainen, VTT Technical Research Centre, Finland
João Pascoal Faria, Faculty of Engineering of University of Porto / INESC TEC, Portugal
Fabrizio Pastore, University of Milano - Bicocca, Italy
Kunal Patel, Ingenuity Systems, USA
Óscar Pereira, Instituto de Telecomunicacoes - University of Aveiro, Portugal
Willy Picard, Poznań University of Economics, Poland
Jose R. Pires Manso, University of Beira Interior, Portugal
Sören Pirk, Universität Konstanz, Germany
Meikel Poess, Oracle Corporation, USA
Thomas E. Potok, Oak Ridge National Laboratory, USA
Christian Prehofer, Fraunhofer-Einrichtung für Systeme der Kommunikationstechnik ESK, Germany
Ela Pustułka-Hunt, Bundesamt für Statistik, Neuchâtel, Switzerland
Mengyu Qiao, South Dakota School of Mines and Technology, USA
Kornelije Rabuzin, University of Zagreb, Croatia

J. Javier Rainer Granados, Universidad Politécnica de Madrid, Spain
Muthu Ramachandran, Leeds Metropolitan University, UK
Thurasamy Ramayah, Universiti Sains Malaysia, Malaysia
Prakash Ranganathan, University of North Dakota, USA
José Raúl Romero, University of Córdoba, Spain
Henrique Rebêlo, Federal University of Pernambuco, Brazil
Hassan Reza, UND Aerospace, USA
Elvinia Riccobene, Università degli Studi di Milano, Italy
Daniel Riesco, Universidad Nacional de San Luis, Argentina
Mathieu Roche, LIRMM / CNRS / Univ. Montpellier 2, France
José Rouillard, University of Lille, France
Siegfried Rouvrais, TELECOM Bretagne, France
Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance, Germany
Djamel Sadok, Universidade Federal de Pernambuco, Brazil
Ismael Sanz, Universitat Jaume I, Spain
M. Saravanan, Ericsson India Pvt. Ltd -Tamil Nadu, India
Idrissa Sarr, University of Cheikh Anta Diop, Dakar, Senegal / University of Quebec, Canada
Patrizia Scandurra, University of Bergamo, Italy
Giuseppe Scanniello, Università degli Studi della Basilicata, Italy
Daniel Schall, Vienna University of Technology, Austria
Rainer Schmidt, Munich University of Applied Sciences, Germany
Cristina Seceleanu, Mälardalen University, Sweden
Sebastian Senge, TU Dortmund, Germany
Isabel Seruca, Universidade Portucalense - Porto, Portugal
Kewei Sha, Oklahoma City University, USA
Simeon Simoff, University of Western Sydney, Australia
Jacques Simonin, Institut Telecom / Telecom Bretagne, France
Cosmin Stoica Spahiu, University of Craiova, Romania
George Spanoudakis, City University London, UK
Lena Strömbäck, SMHI, Sweden
Osamu Takaki, Japan Advanced Institute of Science and Technology, Japan
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Wasif Tanveer, University of Engineering & Technology - Lahore, Pakistan
Ergin Tari, Istanbul Technical University, Turkey
Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Jean-Claude Thill, Univ. of North Carolina at Charlotte, USA
Pierre Tiako, Langston University, USA
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Guglielmo Trentin, National Research Council, Italy
Dragos Truscan, Åbo Akademi University, Finland
Chrisa Tsinaraki, Technical University of Crete, Greece
Roland Ukor, FirstLinq Limited, UK
Torsten Ullrich, Fraunhofer Austria Research GmbH, Austria
José Valente de Oliveira, Universidade do Algarve, Portugal

Dieter Van Nuffel, University of Antwerp, Belgium
Shirshu Varma, Indian Institute of Information Technology, Allahabad, India
Konstantina Vassilopoulou, Harokopio University of Athens, Greece
Miroslav Velev, Aries Design Automation, USA
Tanja E. J. Vos, Universidad Politécnica de Valencia, Spain
Krzysztof Walczak, Poznan University of Economics, Poland
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Wojciech Wiza, Poznan University of Economics, Poland
Martin Wojtczyk, Technische Universität München, Germany
Hao Wu, School of Information Science and Engineering, Yunnan University, China
Mudasser F. Wyne, National University, USA
Zhengchuan Xu, Fudan University, P.R.China
Yiping Yao, National University of Defense Technology, Changsha, Hunan, China
Stoyan Yordanov Garbatov, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, INESC-ID, Portugal
Weihai Yu, University of Tromsø, Norway
Wenbing Zhao, Cleveland State University, USA
Hong Zhu, Oxford Brookes University, UK
Qiang Zhu, The University of Michigan - Dearborn, USA

CONTENTS

pages: 288 - 308

A Visual Language for the Modelling of Service Delivery Processes to Support Business Processes Management

Eunji Lee, SINTEF ICT, Norway
Amela Karahasanović, SINTEF ICT, Norway
Ragnhild Halvorsrud, SINTEF ICT, Norway

pages: 309 - 326

Reconfigurable Equiplets Operating System - A Hybrid Architecture to Combine Flexibility and Performance for Manufacturing

Daniel Telgen, HU University of Applied Sciences Utrecht, The Netherlands
Erik Puik, HU University of Applied Sciences Utrecht, The Netherlands
Leo van Moergestel, HU University of Applied Sciences Utrecht, The Netherlands
Tommas Bakker, HU University of Applied Sciences Utrecht, The Netherlands
John-Jules Meyer, Utrecht University, The Netherlands

pages: 327 - 338

On-Screen Point-of-Regard Estimation Under Natural Head Movement for a Computer with Integrated Webcam

Stefania Cristina, University of Malta, Malta
Kenneth P. Camilleri, University of Malta, Malta

pages: 339 - 349

A Framework for Ensuring Non-Duplication of Features in Software Product Lines

Amal Khtira, ENSIAS, Mohammed V University, Morocco
Anissa Benlarabi, ENSIAS, Mohammed V University, Morocco
Bouchra El Asri, ENSIAS, Mohammed V University, Morocco

pages: 350 - 360

User-Centered Design of a COPD Remote Monitoring Application- Experiences from the EU-project United4Health

Berglind Smaradottir, University of Agder, Norway
Martin Gerdes, University of Agder, Norway
Rune Fensli, University of Agder, Norway
Santiago Martinez, University of Agder, Norway

pages: 361 - 376

Cognostics and Knowledge Used With Dynamical Processing

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster (WWU) and Leibniz Universität Hannover and HLRN, Germany

pages: 377 - 386

Qualitative Comparison of Geocoding Systems using OpenStreetMap Data

Konstantin Clemens, Technische Universität Berlin, Germany

pages: 387 - 397

Conditions Necessary for Visibility and Sensors Placement in Urban Environments Using Genetic Algorithms

Oren Gal, Technion - Israel Institute of Technology, Israel

Yerach Doytsher, Technion - Israel Institute of Technology, Israel

pages: 398 - 409

Study The Throughput Outcome Of Desktop Cloud Systems Using DesktopCloudSim Tool

Abdulelah Alwabel, University of Southampton, United Kingdom

Robert Walters, University of Southampton, United Kingdom

Gary Wills, University of Southampton, United Kingdom

pages: 410 - 419

Semantic Indexing based on Focus of Attention Extended by Weakly Supervised Learning

Kimiaki Shirahama, Pattern Recognition Group, University of Siegen, Germany

Tadashi Matsumura, Graduate School of System Informatics, Kobe University, Japan

Marcin Grzegorzec, Pattern Recognition Group, University of Siegen, Germany

Kuniaki Uehara, Graduate School of System Informatics, Kobe University, Japan

pages: 420 - 434

A Novel Taxonomy of Deployment Patterns for Cloud-hosted Applications: A Case Study of Global Software Development (GSD) Tools and Processes

Laud Ochei, Robert Gordon University, United Kingdom

Andrei Petrovski, Robert Gordon University, United Kingdom

Julian Bass, University of Salford, United Kingdom

pages: 435 - 449

Query Interpretation – an Application of Semiotics in Image Retrieval

Maaïke H.T. de Boer, TNO and Radboud University, The Netherlands

Paul Brandt, TNO and Eindhoven University of Technology, The Netherlands

Maya Sappelli, TNO and Radboud University, The Netherlands

Laura M. Daniele, TNO, The Netherlands

Klamer Schutte, TNO, The Netherlands

Wessel Kraaij, TNO and Radboud University, The Netherlands

pages: 450 - 456

Recognition of Human Faces in the Presence of Incomplete Information

Soodeh Nikan, University of Windsor, Canada

Majid Ahmadi, University of Windsor, Canada

pages: 457 - 466

Abstraction Layer Based Service Clusters Providing Low Network Update Costs for

Virtualized Data Centers

Ali Kashif Bashir, Osaka University, Japan

Yuichi Ohsita, Osaka University, Japan

Masayuki Murata, Osaka University, Japan

pages: 467 - 480

Looking for the Best, but not too Many of Them: Multi-Level and Top-k Skylines

Timotheus Preisinger, DEVnet Holding GmbH, Germany

Markus Endres, University of Augsburg, Germany

pages: 481 - 490

Efficient Selection of Representative Combinations of Objects from Large Database

Md. Anisuzzaman Siddique, Hiroshima University, Japan

Asif Zaman, Hiroshima University, Japan

Yasuhiko Morimoto, Hiroshima University, Japan

pages: 491 - 502

Cloud Computing System Based on a DHT Structured Using an Hyperbolic Tree

Telesphore Tiendrebeogo, Polytechnic University of Bobo, Burkina Faso

Oumarou Sié, University of Ouagadougou, Burkina Faso

A Visual Language for the Modelling of Service Delivery Processes to Support Business Processes Management

Eunji Lee, Amela Karahasanović, and Ragnhild Halvorsrud

SINTEF ICT

Oslo, Norway

{eunji.lee; amela.karahasanovic; ragnhild.halvorsrud}@sintef.no

Abstract—Business process management aims to align the business processes of an organisation with customers' needs. Such alignment is of particular importance for services and requires a good understanding of the interactions among the actors involved. Although several process modelling languages and a service design technique called 'service blueprint' provide good support for modelling of service delivery processes, the actual execution of service and networked interactions among actors seem to have not been sufficiently considered. To overcome these limitations, we developed the Service Journey Modelling Language. Each version of the language has been evaluated, and the results were used as input for the next version. Results from our case studies show that our language might supplement the management of business processes for services by aligning business processes with customer perspectives and by supporting depiction of the actual service journeys with networked interactions of involved actors using appropriate visual representations.

Keywords—business process management; service; process modelling; service design; visual language.

I. INTRODUCTION

This paper is an extended version of 'Can Business Process Management Benefit from Service Journey Modelling Language?' that was published in the proceedings of the Eighth International Conference on Software Engineering Advances [1]. While the original paper investigated if and how a visual language that presented the customer journey through a service might be useful for aligning the business processes of service providers with the expectations of customers, this paper extends the scope, investigating how such a language can support the modelling of a service delivery process characterised by many steps, actors, and interactions among them. In this paper, we distinguish services from products. Services have characteristics that are intangible, extended in time, and delivered across various interfaces.

The importance of services in businesses has grown worldwide [2]. The same trend can be seen in the European Union (EU), where services generate approximately 70% of the EU's GDP and employment [3].

Business Process Management (BPM) is a systematic approach to support the design, enactment, management, and analysis of operational business processes [4]. It defines business processes, shows the interactions among them, and models the organisational structure [5]. The core concepts in

BPM are efficiency and adaptability. BPM is a collaborative effort that involves people and their use of systems in an organisation in pursuit of the organisation's goals. The main goal of BPM is to increase the productivity and efficiency of the organisation's work.

According to Verner [5], there is a lack of techniques for designing, analysing, and simulating business processes. She argued that process modelling languages are the key elements for solving this problem [5]. The importance of service in our economy emphasises the need for a good understanding of BPM in the context of services.

Several process modelling languages have been used in practice. Basic flowchart, UML activity diagram, and Business Process Modelling Notation (BPMN) are some examples. However, the customer's point of view does not seem to be included in most of these languages.

Service blueprint, a process-based technique from service design, has been developed to illustrate the customer's point of view, but the interactive and dynamic nature of service delivery processes are difficult to capture in the service blueprint model [6]. There is a lack of service design techniques in which the actual execution of a service, as opposed to a presumed or expected execution, can be represented or described.

Moreover, most existing tools for business process modelling and service design are rooted in a dyadic relationship (one customer and one provider); whereas today's fragmented service models have other relationships (one customer and many service providers or actors due to outsourcing, specialisation etc.). Tax et al. introduced the concept of a service delivery network (SDN) and defined it as 'two or more organisations that, in the eyes of the customer, are responsible for the provision of a connected overall service experience' [7]. This implies that there is a need for description formats that take a network perspective rather than a dyadic perspective.

In addition, there is a need for a process modelling language with sufficient graphical capacity to support the description of service delivery processes. Service blueprint requires greater visualisation capacity to include more information in a service delivery process [6].

In short, we believe that service providers need an appropriate process modelling language that i) takes into account the service customer's point of view, ii) enables the presentation of the actual execution of the service delivery process, iii) considers the actor network perspective of service provision and consumption in the description format,

and iv) has rich enough graphical capacity. We believe that such a language will eventually contribute to supplement BPM for services.

We developed a visual domain specific language, called Service Journey Modelling Language (SJML). SJML aims to support the analysis of existing services and the design of new services [8]. SJML contains terminology (semantical attributes), graphical attributes (symbols), syntactical attributes, and grammar (rules). To evaluate the very first version of the language, we organised a workshop with the staff of a university library, focusing on describing and redesigning existing services [1]. The language has been further developed by project members through discussions and workshops, and it has been evaluated in two case studies involving an eMarketing company and an energy providing company [8].

This paper extends our previous work by investigating how such a language can support the modelling of service delivery processes characterised by many actors and steps, and how to describe highly complex service delivery processes appropriately. To do this, we conducted a case study with a referral process. A referral process is the process of transferring the care of a patient from one medical expert to another [9]. We used our language to describe an electronic referral process in a Norwegian hospital. Several challenges were identified. To address them, we further extended the language.

The rest of this paper is organised as follows: Section II describes the theoretical background and related work in the areas of business process modelling and service design techniques for process modelling. Section III presents the overall requirements for SJML and describes a case study with SJML v1.0. Section IV shows the development process of SJML v2.0 with details of a case study. In Section V, we present and discuss the results from our case studies. Section VI discusses the limitations of this study. Section VII concludes this paper and proposes future work.

II. RELATED WORK

Our work spans quite a number of different disciplines, and we have chosen to focus on the following: service design, information visualisation, communication theory, and modelling languages. In subsection A, we introduce the theoretical background of the abovementioned four disciplines that are related to our research. In subsection B, we describe some existing process modelling languages that demonstrate the sequence of steps in a business process. In subsection C, we introduce a service design technique that has been used to illustrate service delivery processes. In subsection D, we discuss the previous mentioned process modelling languages and the service design technique, concluding that there is a need to develop a new language.

A. Theoretical Background

In this subsection, we introduce the theoretical background of our research. Relevant knowledge from the areas of service design, information visualisation, communication theory, and modelling languages is explained.

1) Service Design Phases

Services are designed in an iterative manner [10][11][12]. The design of services consists of several phases. Dubberly and Evenson introduced five steps (observe, reflect, make, socialise, and implement) [12], and Stickdorn and Schneider introduced four phases (exploration, creation, reflection, and implementation) [13]. The service delivery processes are specified prior to implementation. However, the 'fuzzy front end' in service design and development has been problematic; thus, there is a need for a methodologically structured way to present the service delivery processes [14].

We aim to research how to assist the activities of specifying, describing and analysing the service delivery processes to support the service design phases. Information visualisation, communication theory, and modelling languages have formed a theoretical basis for our research.

2) Information Visualisation

Visualisation originally meant forming a mental picture of something [15]. Information visualisation is a visual representation of abstracted data and concepts that include numerical and non-numerical data. A good example of information visualisation is weather forecasting. Moreover, information visualisation increases human cognition [16] by providing (interactive) visual representation of data or concepts. The term has slightly shifted in meaning from being an internal construct of the mind to an external artefact supporting understanding and decision-making.

Pictures are routinely used to present information in different fields such as medicine, architecture, geography, statistics and design. One of the advantages of using pictures in this way is that a large amount of complex information can be quickly interpreted if presented effectively. Information visualisation helps people easily understand complex information and changes over time that would otherwise be difficult to comprehend [17]. Extensive research suggests that in many application domains like reverse engineering, software restructuring, and information retrieval, information visualisation may improve efficiency, accuracy, and user satisfaction when solving complex tasks [18][19].

By adopting Ware's advantages of information visualisations [20] and Bitner et al.'s gaps model of service quality [21], some expected advantages of information visualisation in the context of describing service delivery process could be:

- Important information about the described service delivery process would be easy to understand.
- Gaps between the expected or prescribed service delivery process and the actually delivered service process will be emphasised.
- Identification of problem areas in need of improvement in a service delivery process will be facilitated.

3) Communication Theory

Communication is the process of sending and receiving messages or transferring information from a sender to a receiver. It is often described in accordance with the sender/receiver-model developed by Shannon and Weaver [22]. The model consists of eight key elements required for communication: source, encoder, message, channel, decoder, receiver, noise and feedback, see Figure 1.

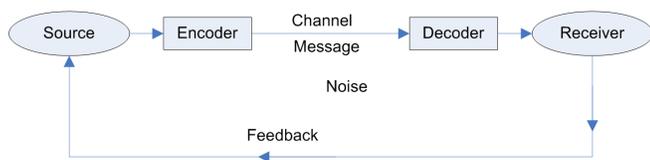


Figure 1. Shannon-Weaver's model of communication

The source of communication is the initiator – an individual or group - with a message or specific intention to start the communication process. The message is communicated from the source to an intended receiver. The encoder provides the format of the message. The channel is the mediator of the message and makes sure it is transmitted to the receiver. To illustrate, the channel may be verbal, written, or electronic. Noise may disturb the transmission during the communication process. It could be physical noise, like in actual sound transmission. However, it can also be semantic noise, for example if the receiver cannot understand the message. Before reaching the destination, the message must be interpreted or decoded. Feedback indicates whether the message has been received and interpreted accurately. It ensures that the source knows whether the communication was successful.

Although this linear model was developed in a time with no multimedia and interactive environments, it is still a widely accepted model when drilling down a communication process in its bits and bytes. However, the model lacks the relational and cultural factors, so it appeals more towards the technological perspectives such as in the development of information retrieval systems. Craig

describes communication theory as a non-coherent field, distinguishing seven different traditions or approaches that contribute significantly with their characteristic ways of defining communication and related challenges [23]. These seven traditions are referred to as rhetorical (the practical art of discourse), semiotic (inter-subjective mediation by signs), phenomenological (experience of otherness and dialogue), cybernetic (information processing), socio-psychological (expression, interaction, and influence), socio-cultural (production of social order), and critical (discursive reflection) traditions.

Semiotics is the study of signs, sign processes, and how signs take part in communication. Although semiotics and communication have a wide intersection in terms of the general phenomena they investigate, their perspectives on the underlying theory of communication are very different [24]. One semiotic interpretation of the Shannon-Weaver communication model was proposed by Jakobson [25]. This model consists of six related elements: context, sender, receiver, message, code, and channel. The sender and receiver may alternate their roles, and the message has a context.

Adopting Jakobson's semiotic interpretations of the Shannon-Weaver communication model [24] might be useful when applying information visualisation in service delivery process description. Considering the six elements of the model might be beneficial for modelling of service delivery processes.

4) Modelling Languages

Modelling languages are artificial languages that show systems or information in a structure. A modelling language is composed of a description of its semantics and syntax [26]. Modelling languages define models that help us to understand how things are and how things behave. People model a language in order to communicate in a more effective and accurate manner and to exchange messages quickly and precisely without extraneous information.

A domain specific language (DSL) is defined as a tailored language for particular application domain [27]. DSLs have evolved based on the needs for more effective communication among people in specific domains. The London metro map, which was pioneered by Beck in 1931 [28], is a good example of DSL. Compared to general purpose languages, DSLs offer improved expressiveness and ease of use [27].

A domain specific modelling language (DSML) allows description of models in a specific area. Thus, a good DSML can support better cooperation in that specific area.

Developing a DSML might be a good solution for describing service delivery processes in order to support cooperation in service design. In addition, the DSML would support understanding information in service delivery processes and facilitate communication among the individuals involved in service design.

B. Process Modelling Languages

In this subsection, we present several depictions of a service delivery process created using some existing modelling languages in BPM. The applied languages are process modelling languages focused on representing the sequence of steps in BPM.

We used a referral process as an example to examine different techniques. There are essentially two types of referrals in Norway: external and internal. An external referral indicates a referral from a primary care provider (general practitioner, GP) to a secondary care provider (specialist). An internal referral indicates a referral from a department in a hospital in which a patient is admitted to another department in the same hospital or to another hospital in the same health region.

In our case study, we focused on the external referral process. The general activities of an external referral process in Norway are as follows. First, a patient referral letter (in paper or electronic form) from a GP is received in a hospital. Second, the letter is assessed by a medical expert. Third, depending on the result of the assessment, the patient gets an appointment with a specialist or is required to wait. Finally, the patient receives treatment, and the specialist sends an epicrisis, which is an analytical summary of a medical case history, to the GP who sent the referral letter. However, the referral process may vary among clinics, hospitals and regions.

In this subsection, an example of the initial part of an external referral process to an outpatient clinic in a Norwegian hospital is described using different process modelling languages. The details of our case study are described in Section IV. A.

Our example begins with a post document centre forwarding a referral letter. A specialist then screens the referral and, if another specialist group needs to assess the referral, returns it to the post document centre. Otherwise, the specialist assesses the referral. Depending on the content of the referral and the patient case history, the specialist assesses whether the patient needs prioritised medical assistance. If so, a nurse arranges an appointment for the patient. If not, the information is stored in an electronic health record (EHR) system for later scheduling.

1) Flowchart

The first process modelling language that was examined is a flowchart, which is a type of diagram that describes a process or workflow. The steps of the process are illustrated with various shapes and arrows that constitute the sequence of activities. Flowcharts are used in various fields to analyse, design, document or manage a process or program [29].

Figure 2 shows a basic flowchart describing the initial part of the previously mentioned referral process. The rounded rectangle on the top indicates the start of the

process. The rectangle boxes indicate activities. The diamonds indicate decision points, and the arrows show the activity flow.

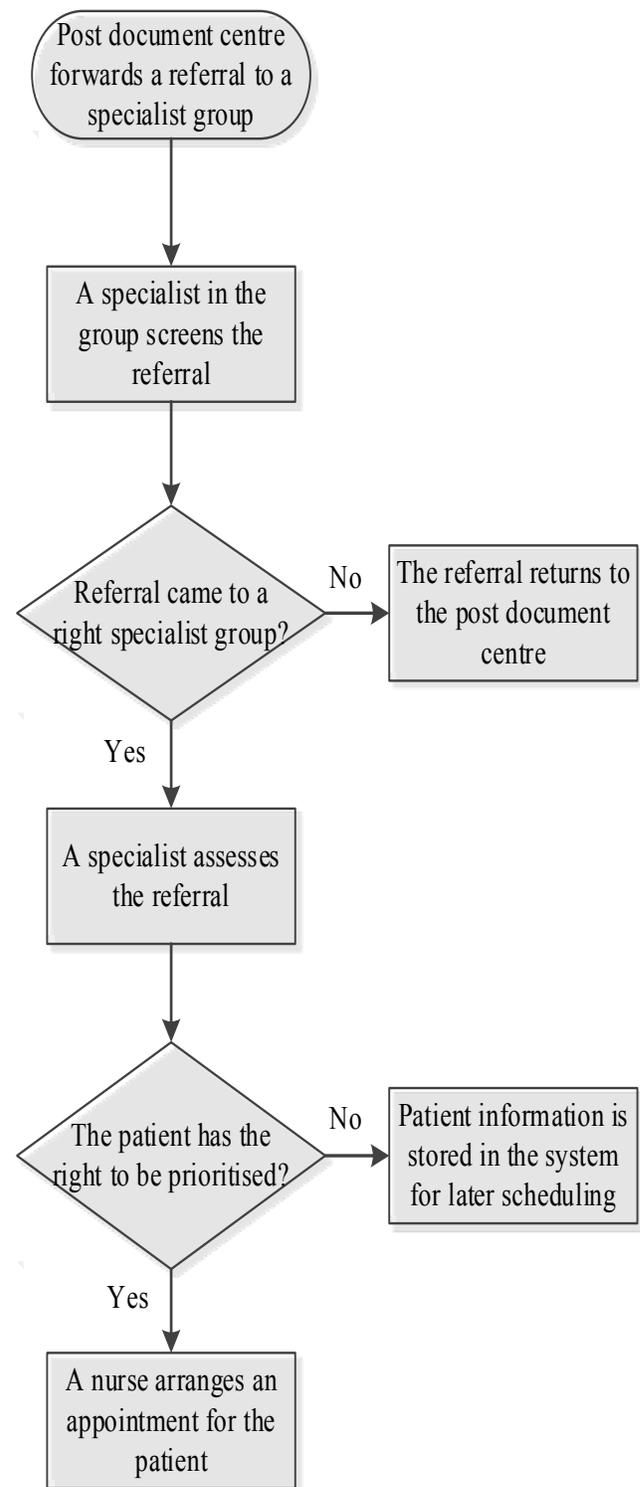


Figure 2. An initial part of a referral process illustrated by a basic flowchart

2) Unified Modelling Language Swimlane Activity Diagram

Unified Modelling Language (UML) is defined as ‘a graphical language for visualising, specifying, constructing, and documenting the artefacts of a software-intensive system’ [30]. UML consists of visual notations and the rules for the use. It includes many types of diagrams, such as class diagrams, use case diagrams, activity diagrams, etc. Eriksson and Penker claimed that the most important UML diagram for business process modelling is the activity diagram [31].

UML swimlane activity diagram is a type of activity diagram that can divide activities by roles or locations. Figure 3 shows the initial part of the referral process (the same part as Figure 2), which is described using a role-based swimlane activity diagram. Each swimlane (column) shows a different role participating in the process. The black circle on the top indicates the starting point. The diamonds show the decision points. The rectangular boxes indicate the activities, and the arrows show the transition flow from one activity to another.

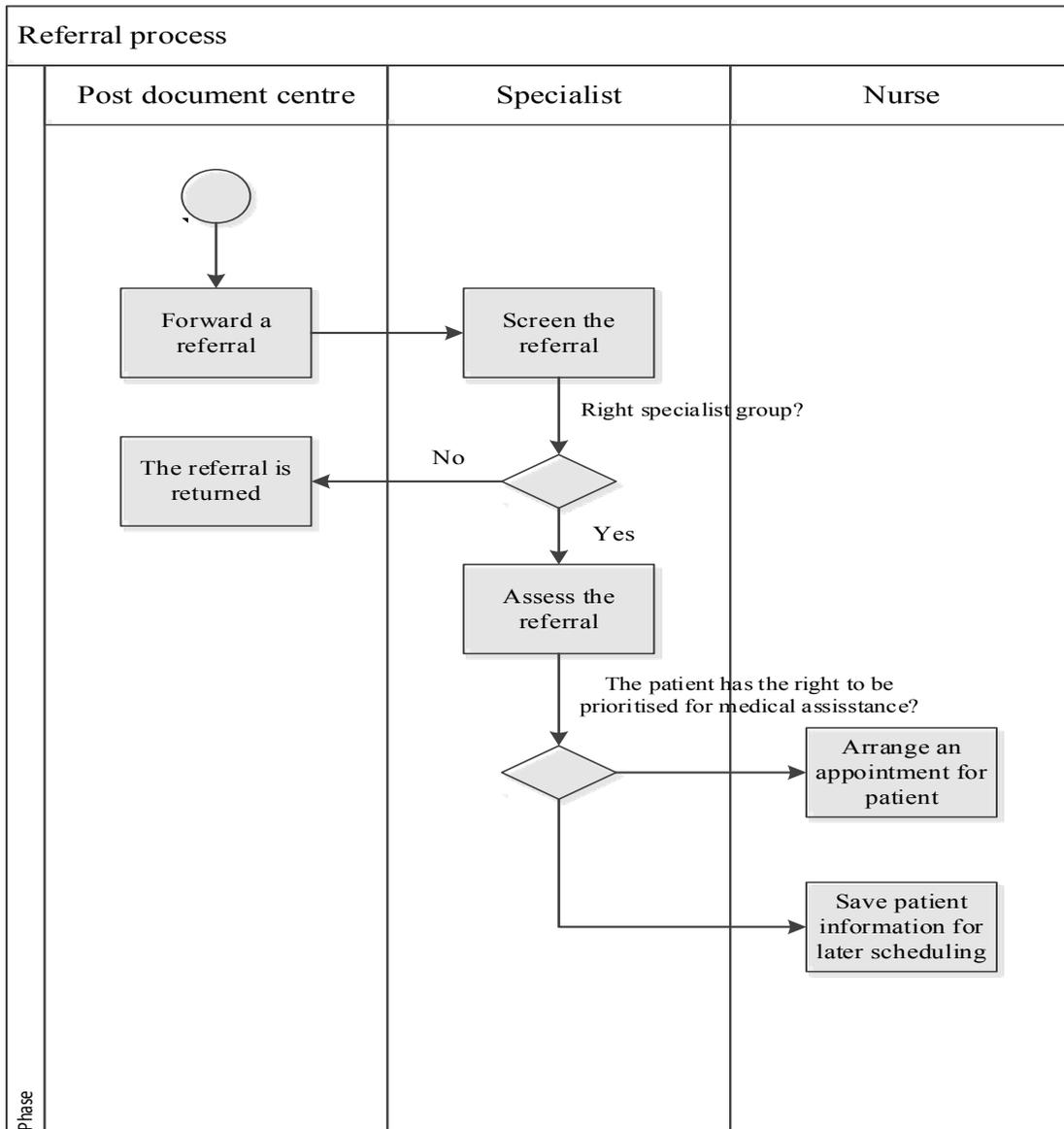


Figure 3. An initial part of a referral process described with a UML swimlane activity diagram.

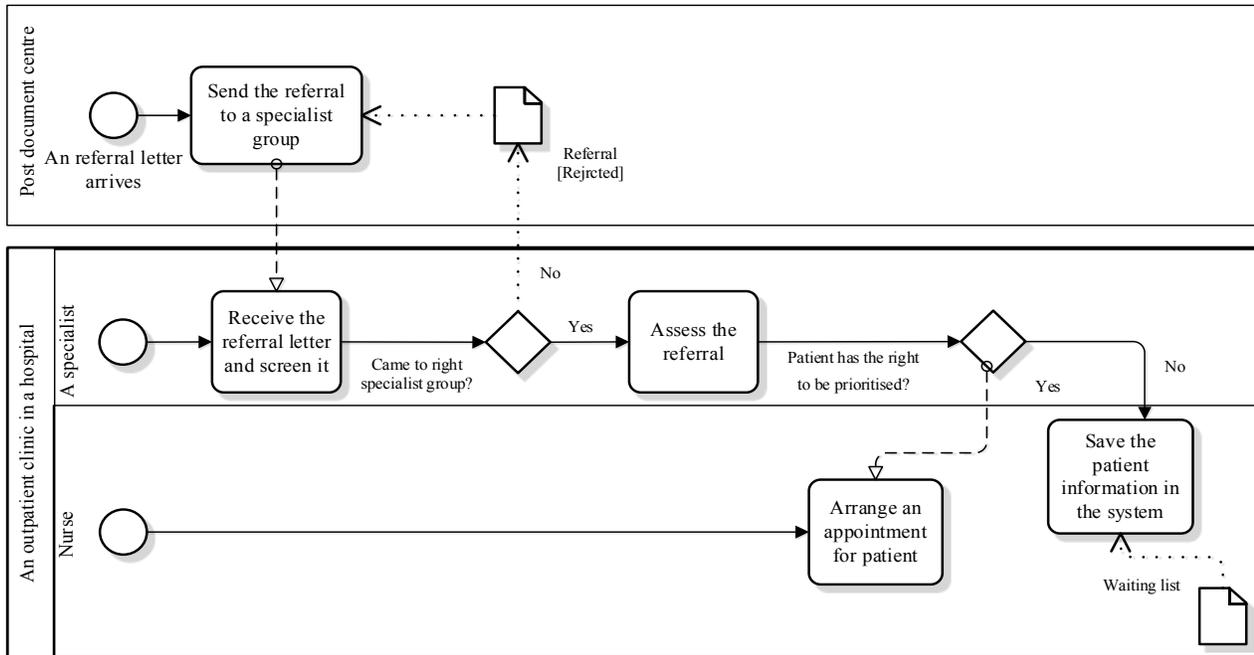


Figure 4. An initial part of a referral process illustrated by a BPMN diagram.

3) Business Process Modelling Notation

Business Process Modelling Notation (BPMN) is considered to be an ‘understandable graphical notation for all business users’, ranging from business analysts and technical developers to business managers, which can reduce the gap between business process design and process implementation [32]. It describes the steps in a business process using a flowchart that defines business process workflows [33].

Figure 4 shows an example of the use of BPMN for the initial part of the referral process. Similar to UML swimlane activity diagram, BPMN allows for description of the swimlane, activity, decision point, event, data object and pool, and provides three types of connectors: sequence flow, message flow, and association. However, BPMN takes a process-oriented approach while UML takes an object-oriented approach, and BPMN is designed to be more suitable for a business process domain so that it can better support BPM. In Figure 4, two locations are shown, the post document centre and the outpatient clinic in the hospital. In our example, there are two actors in the outpatient clinic: a nurse and specialist.

C. Technique in Service Design

In this subsection, we introduce a DSML equivalent service design technique that describes service delivery processes.

1) Service Blueprint

Service blueprint is a service design technique introduced by Shostack that has been widely used in service management and marketing [34]. It shows the series of service actions and the time flows that are related to the roles of the actors involved in a service delivery process.

The actions that customers take as part of the service delivery process are separated from the actions of the contact person by the line of interaction. The line of visibility differentiates the actions of an on-stage/visible contact person (actions that are seen by the customer) from the actions of a back-stage/invisible contact person (actions that are not seen by the customer). Support processes, which include all the activities carried out by individuals or units in the company that are not contact persons, are separated from the actions of the contact person by the line of internal interaction. Physical evidences refer the media that customers come in contact with when they perform their actions.

A service blueprint enables managers to have an overview of an entire process and provides useful information for the development and evaluation of new services [14]. Figure 5 shows the initial part of the referral process using the extended service blueprint introduced by Bitner et al. [14]. In Figure 5, the customer is a specialist in the hospital.

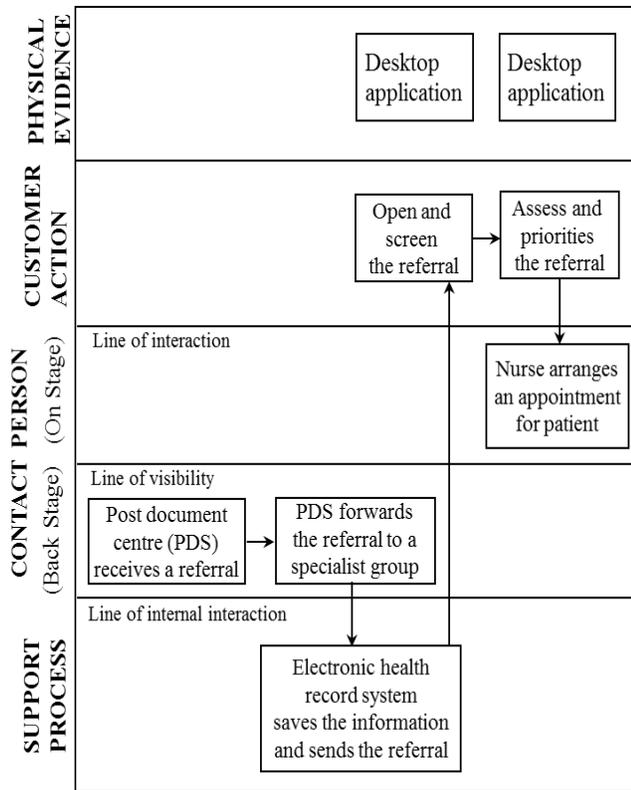


Figure 5. An initial part of a referral process drawn by a service blueprint

D. Summary

Flowcharts help us understand the overall processes [35], but the lack of visual notations to represent actors and their interactions might hinder the reader’s ability to clearly understand the service delivery process. UML swimlane activity diagram shows the sequence of activities with clear role definitions [35], but it still does not have any visual differentiator for actions and interactions. BPMN is an industry standard for process modelling and has visual notations for event, data object and various types of connectors (sequence flow, message flow and association). Nevertheless, it supports no visual differentiator for actions and interactions. Recker et al. argued that BPMN still requires more symbols with extended expressiveness to add sufficient consistency for making the models fit for use [36]. Afrasiabi et al. claimed that there is a lack of standard graphical notation for business process modelling languages [37]. The same applies to service blueprint. There is no standard service blueprint with common terminology and visual language [38]. Therefore, to include more information about a service delivery process, the service blueprint format must improve its visualisation capacity [6].

Flowchart, UML swimlane diagram, and BPMN illustrate only the expected service delivery processes. They were developed to be used to prescribe planned processes. Service blueprint allows us to see the technical parts of the service delivery process that customers cannot see. However, it often only supports description of the expected service delivery processes, not the actually executed ones.

In addition, service blueprint does not take a network view of services. Service blueprint does not support describing complex service structures that involve relationships among actors and events [6]. UML swimlane activity diagram, and BPMN can support the description of networked interactions among actors, but flowchart and service blueprint might not be able to do so.

Table I summarises the different process modelling languages and the service design technique for BPM that have been described. The second column from the left indicates the purpose for using the language or technique. The third column indicates whether the language or technique considers a service providers’ and/or customers’ perspective. The fourth column indicates whether the language or technique supports the presentation of actually executed service delivery processes or the presentation of prescribed service delivery processes. The fifth column shows what kind of customer and service provider relationship the language or technique supports.

Flowchart is used to describe a sequence of interactions, while UML swimlane activity diagram is used to illustrate a sequence of interactions and data. BPMN is used to demonstrate a sequence of interactions and data for business purpose, and service blueprint is used to draw a sequence of interactions for service design. The presented languages and technique mainly consider the service providers’ point of view. Flowchart, UML swimlane activity diagram, BPMN, and service blueprint can support the description of only the expected or prescribed execution of a service. UML swimlane activity diagram and BPMN can support the illustration of networked relationships among the customer, service provider, and third party, but flowchart and service blueprint support the illustration of the dyadic relationships.

Therefore, there is a need for a process modelling language that includes service providers’ and customers’ perspectives, and can support the presentation of the actual execution of a service and the networked relationships among actors. A service domain specific process modelling language that satisfies these aspects will be introduced in Section III. This process modelling language was designed to address the weaknesses of the existing languages and technique by adopting information visualisation and communication theory. Thus, it aims to show the interactions among different actors in an easily understandable manner. In addition, the language was designed to have a rich enough graphical capacity to properly illustrate the service delivery process.

TABLE I. PROCESS MODELLING LANGUAGES AND SERVICE BLUEPRINT

Language/technique	Purpose of the use	Perspective of the presentation	Presentation type	Customer and service provider relationship
Flowchart	Sequencing of interactions	Service provider-oriented	Prescriptive	Dyadic relationship
UML swimlane activity diagram	Sequencing of interactions and data	Service provider-oriented	Prescriptive	Networked relationship
BPMN	Sequencing of interactions and data for business purpose	Service provider-oriented	Prescriptive	Networked relationship
Service blueprint	Sequencing of interactions for service design	Service provider-oriented (back-stage oriented)	Mainly prescriptive	Dyadic relationship

III. SETTING THE SCENE: REQUIREMENTS AND EARLY VERSIONS OF THE SERVICE JOURNEY MODELLING LANGUAGE

This section presents our first steps in developing the modelling language. Subsection A describes the overall requirements and the development procedure, and subsection B describes the elements of SJML v1.0. Subsection C describes the initial feedback on SJML v1.0 while subsection D briefly describes SJML v1.1.

A. Overall Requirements and Development Process

We wanted to develop a language that supports specification, description and analysis of the service delivery processes for service design, as described in Section II.A. Our language is based on a customer journey mapping approach, which is introduced below.

Customer journey maps (CJM) are widely used in service design to visualise service delivery processes from the end user's perspective. CJM allows description of the details of service interactions and associated emotions in a highly accessible manner [13]. CJM is often structured around touchpoints and is confined to a certain period of time. A touchpoint corresponds to an interaction between a customer and a service provider, but may also denote a communication channel between the customer and service provider. CJM is one of the most commonly used visualisation techniques within service design, where it utilised to obtain an overview of the customer's service consuming process. It communicates customer insight [39], which in turn can be used to identify problem areas and opportunities for innovation [13]. Using CJM, one can easily compare a service with its competitors [13].

We expanded the concept of CJM by addressing the requirements that are explained in the next paragraph. The research was conducted according to the design science's three phases (problem analysis, solution design, and evaluation) in an iterative manner to create artefacts, the language and knowledge in the research domain.

The language we developed was designed to:

- Strengthen customer orientation and facilitate collaboration among all actors through a common vocabulary and extensive use of visualisation

- Support service design and development by means of a precise language for specifying actors, interactions among them, the timing of their interactions, and the communication channels used
- Support the analysis of services to check consistency; in particular, it facilitates the monitoring of service execution by comparing it to the expected or prescribed service delivery process

Table II provides an overview of the versions of SJML. The first column on the left shows the version. The second column presents the elements of the version. The third column describes the type of the study conducted for evaluation and further development. The fourth column shows whether the version was evaluated with existing or new services. The fifth column presents the degree of complexity of the service(s) it portrayed. The sixth column shows the place(s) where the version is described.

A touchpoint is a point of interaction between a customer and a service provider, and it is a central element of our language. SJML v1.0 consisted of basic elements (touchpoint, channel, actor, and status) to describe a sequence of touchpoints for expected journeys (sequential view). SJML v1.1 added new diagram elements and a diagram type. Visual notations for action, decision point, concurrency, and un-ordered sequence were added, and a deviation view was introduced to describe the actual journey. SJML v2.0 added additional diagram elements and adjusted some elements. Diagram elements for system activity, touchpoint description, and time flow were added, and visual notations for touchpoint, action, actor, initiator, and recipient were modified. A new diagram type (swimlane view) was introduced to illustrate the service journeys of several actors to support the description of a networked perspective of the service journeys. The evaluations were performed by describing services using the three versions of our language. The earliest version of the language was evaluated by describing simple services (book loan services at a library). The results are described in the original paper [1]. More complicated services (customer enrolment services) from our research partner companies were described using SJML v1.1, and the findings were reported in a conference paper [8]. SJML v2.0 was developed and evaluated by describing an even more complicated service (a healthcare service) that involves many actors and the interactions among them. The results and findings are described in Section IV.

TABLE II. CHARACTERISTICS OF THE DIFFERENT VERSIONS OF SJML

Version	Language composition	Evaluation			Described in
		Type of study	Service type	Complexity	
SJML v1.0	<i>Basic attributes that form a basic diagram type</i> Basic diagram elements (touchpoint, channel, actor, and status) supporting description of a basic diagram type (sequential view)	Preliminary experience with library services	Existing service and new service	Simple service	[1] and Section III. of this paper
SJML v1.1	<i>SJML v1.0, additional attributes, and a new diagram type</i> New diagram elements (action, decision point, concurrency notation, and un-ordered sequence) and a new diagram type (deviation view) were added in SJML v1.0	Two case studies with customer enrolment services	Existing services	More complicated service	[8] and Section III.D. of this paper
SJML v2.0	<i>Adjusted and additional attributes, and a new diagram type</i> Adjusted elements (touchpoint, action, actor, initiator, and recipient), new diagram elements (system activity, touchpoint description, and time flow), and a new diagram type (swimlane view)	A case study with a referral process in healthcare service	Existing service	Even more complicated service	Section IV. of this paper

B. Service Journey Modelling Language v1.0

The first version of SJML (SJML v1.0) consisted of terminology, symbols, graphical elements, visual syntax, and visualisation modes. As this subsection represents the first phase of our research, we wanted to investigate the needs of practitioners when designing new services or improving existing services; what information about customers and their interactions in the service journey is essential to be monitored to align business processes with customers' needs.

Each touchpoint has a symbol inside a circle. The symbol shows a channel or device that is used for the touchpoint. The colour of a touchpoint's boundary indicates the actor who initiates the touchpoint. The boundary style specifies the status of the touchpoint. The elements for SJML v1.0 are detailed below.

- **Touchpoint.** Service journey consists of touchpoints that are described with circles.
- **Channel.** Each touchpoint has a symbol inside a circle. The symbol shows a channel or device that is used for the touchpoint.
- **Actor.** The colour of the touchpoint's boundary indicates the actor who initiates the touchpoint.
- **Status.** The boundary style indicates the execution status of the touchpoint (solid boundary: completed, dashed boundary: missing, and crossed touchpoint: failed).

C. Evaluation of SJML v1.0

To obtain initial feedback on SJML v1.0, we organised a half-day workshop with library staff. The purpose was to investigate SJML v1.0's feasibility and collect requirements for further development. The workshop was arranged for June 2013 as a service design seminar on the premises of the science library at the University of Oslo. It consisted of a lecture about service design and two practical sessions. SJML was presented and evaluated during one session.

The researchers collected data through participatory observation and a plenary discussion. Thematic coding [40] was used to analyse the field notes of the researchers and summarise the results.

The SJML session included a short introduction to the relevant terminology and visual notations, and a presentation of the various exercises to be performed by the participants. Twenty-six individuals participated. Seventeen were library employees, four were students, and five were researchers. The session lasted about 30 minutes and concluded with a group discussion.

Participants were divided into four groups and were asked to construct a customer's service journey of borrowing a paper or an electronic book at the library. One blank circle plus seventeen book-loan service-relevant symbols, which were selected from 32 SJML symbols, were given to each group as a set (Figure 6).



Figure 6. SJML symbols given at the workshop

In this workshop, we observed the use of the touchpoints. Pen and paper were provided to the groups in order to allow them to draw the sequence of the touchpoints. The members of the groups were allowed to create their own touchpoint with the blank circle in the event they do not find the touchpoint they needed. The actor and status attributes of a touchpoint were not adopted for this session.

The first task was to construct the customer journey for borrowing a paper/electronic book (Figure 7). The process of borrowing a book included extension of the loan and finished when the book was returned. The second task was to construct the customer journey for ordering a paper/electronic book that the library did not have. The process included extension of the loan and finished when the book was returned. Participants were asked to make customer journeys for both existing (Figure 7) and desired book loan services.

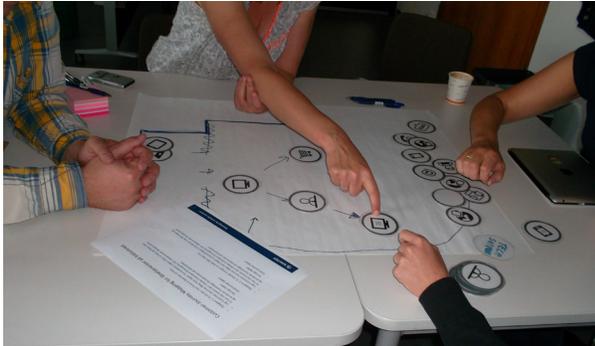


Figure 7. Participants are on the process of making service journeys for an existing book loan service (borrowing of an electronic book)

Results

We observed that the participants did not have any problems understanding SJML and using its symbols. Participants were able to construct and explain their service journeys using SJML. However, some participants were confused about using symbols that looked similar to each other, such as the symbols representing PC and web access via PC. During the feedback discussion, it became evident that the participants missed certain functionalities as directing the flow of the process through conditional gateways or decision points that occur repeatedly during such services. It was also found that more icons would be needed to illustrate the library services more precisely.

D. Service Journey Modelling Language v1.1

Based on the initial feedback on SJML v1.0 and the requirements that were subsequently discovered, we extended the language. Action was added as one of the main elements along with touchpoint, and more touchpoint attributes were added. SJML v1.1's detailed description and our experience with the language can be found in [8]. The elements for SJML v1.1 are detailed below.

- **Touchpoint, channel, actor, and status** are the same as in SJML v1.0.
- **Action.** A service journey consists of touchpoints and actions. An action is an event or activity conducted by a customer or service provider as part of a customer

journey. Unlike a touchpoint, an action does not include any form of communication between the customer, the service provider or a third party.

- **Decision point.** A decision point refers to an expected customer journey being split into 'sub-journeys'. It is illustrated using a square diamond.
- **Concurrency.** In some situations, a touchpoint occurs at the same time as another touchpoint. We call this concurrency and describe it with two separated half-dotted circles. A visual indicator (two clocks and a wave symbol) was introduced to represent the space between the two half-dotted touchpoints.
- **Unordered sequences.** Brackets were introduced to describe unordered sequences of touchpoints.

IV. ADDING COMPLEXITY: FURTHER DEVELOPMENT OF THE LANGUAGE

We wanted our language to be able to describe complex service journeys. Thus, SJML v1.1 was updated to present many actors and their interactions by adapting the swimlane approach. This section describes the use of SJML v2.0 for modelling complex service delivery processes and our experience with it in a hospital case.

A. Case Study with Patient Referral Processes

To identify a set of requirements for a visual language related to modelling the processes of complex services, we conducted a case study involving the modelling of referral processes in health care services. We collected data regarding the referral process from a Norwegian hospital during the autumn of 2013 and used SJML to model the various aspects of the processes.

Due to the number of actors involved, interactions among them, and the number of steps in the process, the referral process is regarded as one of the most complicated services [41]. A referral process is a highly complex process because it often involves many actors (e.g., a patient, a healthcare provider who sent the referral like a GP, administrative personnel such as a receptionist, nurses and medical experts at the receiving facility) and their tasks are interrelated across different organisations, such as a GP's office and a hospital.

The external referral process in the Norwegian hospital where we collected data consists of the following steps: a health secretary receives referral letters and sorts them; a medical expert assesses the referral and makes a decision; a nurse or health secretary follows up the referral, sets a time for a patient visit and notifies the patient; the medical expert meets the patient; and the medical expert communicates with the healthcare provider who referred the patient. Patient-administrative systems are central to the referral process, and the EHR system is one of the core patient-administrative systems used by the healthcare personnel.

B. Data Collection

To understand how the referral process works and what it consists of in practice, we visited a surgical outpatient clinic at a hospital in Norway three times during September and October of 2013. We collected data using observations and semi-structured interviews during our visits. A researcher wrote down field notes during the observations and the interviews with administrative personnel, nurses, and medical experts. We also studied an E-learning's module, documentation, and photos that are relevant to the referral process. The detailed data collection procedure is introduced below.

We first studied the hospital procedures and routines concerning referrals in general using an E-learning module for the EHR system. We then classified the procedures. After that, actual data from the referral processes were collected through observations. These data consisted of three detailed patient histories that were extracted from the EHR system. Photographs of screenshots were taken during the observations. Then, the data were supplemented by semi-structured interviews with a nurse, a medical specialist, and a health secretary. The interviews were audio-recorded. Relevant documents such as patient brochures and documentation explaining the referral process and internal routines were also obtained. In addition, we had telephone meetings with a nurse and asked her for some additional explanations via email.

C. Data Analysis and Further Development of SJML

Reconstruction of the three patient cases formed the core of our analysis. The three patient histories were reconstructed by combining the patient journal with the referral module in the EHR system. A chief nurse anonymised the data from the patient journal and assisted in constructing the patient histories. To process the fragmented data from the interviews and the EHR system, each patient history was compiled and systemised using a preformatted spreadsheet template. The patient histories were organised by date, physical location, and events, and attributes such as initiator, recipient, task description, and communication channel were specified.

The results from the data analyses were used to describe the referral processes using SJML v1.1. We organised workshops in which three researchers sketched different approaches for visualising fragments of the referral process through trial, error, and exploration. When visualising the referral processes, we could also identify requirements for the further development of our visual language. We improved our language on the basis of the requirements through several iterations. Those requirements and our solutions are described as following.

- (1) *Ability to visually describe and distinguish among several actors in a more appropriate way*

Using different colours to distinguish among actors might not be suitable in case many actors are involved. Thus, we needed a new way to describe several actors. Alternative representations were explored through trial and evaluation of various solutions. Finally, swimlanes with actor icons and titles were adopted. This approach seems applicable to account for all the actors involved.

- (2) *Ability to describe and distinguish each actor's touchpoints that connect the actors*

Since we adopted the swimlane approach, there was a need for a new way to describe touchpoints that involved communication among actors. The language should therefore have touchpoints for both the initiator and recipient. We adopted a sender-recipient concept. Each touchpoint was duplicated and placed in both the initiator's and recipient's swimlane so the readers could recognise each actor's journey easily by following each swimlane. The boundary colour of a touchpoint showed if the touchpoint was sent or received by the actor.

- (3) *Ability to distinguish among touchpoints mediated by the same channel, and thus having the same symbols*

Sometimes, several touchpoints occur through the same communication channel. For instance, a customer may receive two SMS messages from the service provider. We proposed a text area to allow for a detailed annotation of the touchpoint. The textual explanation, together with a symbol of the channel inside a touchpoint square box, would facilitate a detailed understanding of the touchpoint.

- (4) *Ability to describe dataflow in the EHR system*

Some information is delivered from sender to receiver through the EHR system, such as when updating and saving information. Therefore, we needed a way to describe dataflow in the EHR system. We proposed adding the EHR system as an actor and using arrows to illustrate dataflow.

- (5) *Ability to distinguish between the workflow of referral and dataflow*

Since we used an arrow for both workflow and dataflow, we needed a way to distinguish between these two types of flow. We proposed using differently shaped arrows for workflow and dataflow. Normal black arrows were adopted to describe workflow while dotted arrows were adopted to describe dataflow.

D. Components for Visualisation

Figure 8 shows SJML v2.0 representing process steps in the form of touchpoint or action. The text area enables annotation concerning the context of the process step. In SJML v2.0, a box was used to represent touchpoints and actions as its shape can contain text more economically than a circle. The colour of the touchpoint boundary indicates the initiator (blue) and recipient (green), and a symbol area carries information about the channel or device mediating the touchpoint. In contrast to touchpoints, actions have no sender or recipient, thus they do not need a colour to distinguish between sender and recipient, nor a symbol to represent the communication channel.



Figure 8. Visual components for our expanded language

Figure 9 presents an excerpt of SJML v2.0, showing two touchpoints in a patient’s journey. On the left is a symbol representing the patient along with a text label. The first touchpoint represents an interaction in which the patient receives a phone call from the hospital (a common procedure when scheduling an urgent appointment). The next touchpoint shows an interaction in which the patient receives the appointment letter with all necessary details about the imminent consultation and a brochure regarding patient rights. Here, the patient is the recipient of both touchpoints, signalled by the green colour. Correspondingly,

the health secretary responsible for the telephone call and the letter is the initiator of both touchpoints. The corresponding touchpoints of the health secretary would appear with blue-coloured boxes in a separate swimlane.

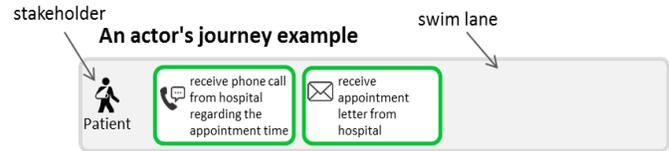


Figure 9. A part of a patient's journey

E. Mapping in a Real Context

Figure 10 shows a part of the real referral process described using SJML v2.0. Arrows indicate the workflow and dataflow. Black arrows represent the workflow in the process while grey dotted arrows represent the dataflow to/from the EHR system. Time flow with time stamps is shown by the black line on the bottom. This example is based on one of the real patient histories explained in subsection D.

First, a GP transferred an electronic referral letter to a hospital for a patient with a lipoma who visited him/her. Second, a health secretary in the document centre of the hospital received, registered and forwarded the referral to a specialist in the hospital. Third, the specialist assessed the received referral and forwarded it to a health secretary in the clinic for follow-up. Fourth, the health secretary checked the received referral and sent a standby letter to the patient, and so on. The entire version of this referral process can be found in Appendix A.

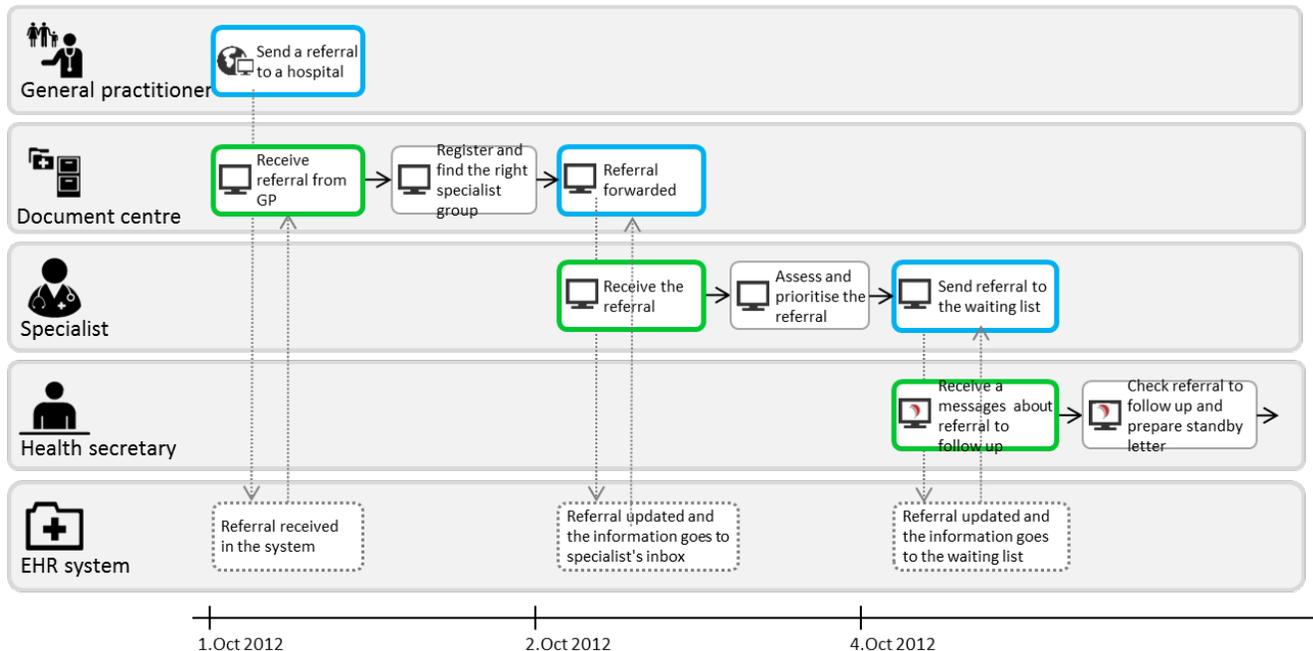


Figure 10. The result of the proposal for extension of the language

F. Service Journey Modelling Language v2.0

SJML v2.0 consists of a sequence of touchpoints, actions, and system activities. The elements of SJML v2.0 are detailed below.

- **Touchpoint.** A touchpoint indicates a point of interaction between two actors. The shape of the touchpoint was changed from a circle to a rounded rectangle since more space was needed for text to add information concerning the touchpoint.
- **Action.** An action indicates an event performed by an actor as part of a customer's journey. It is represented by a rounded grey rectangle.
- **System activity.** This is an activity performed by the EHR system. It is represented by a dotted grey rectangle.

Attributes of touchpoints, including channel, status, initiator, and receiver, and touchpoint description were specified for SJML v2.0.

- **Channel.** The symbol area for channel has been moved to the upper left of a rounded rectangle touchpoint.
- **Status.** The way to describe the status of a touchpoint is the same as in SJML v1.0 and v1.1.
- **Initiator and recipient.** Initiator (rounded rectangle with blue-coloured boundary) indicates an actor who initiates the touchpoint, and recipient (rounded rectangle with green-coloured boundary) indicates an actor who receives the touchpoint. We found through the case study that there could be more than one recipient of a touchpoint in some cases.
- **Touchpoint description.** A text area has been added on the right side of the rectangle to describe the touchpoint in detail.

Graphical notations have been modified for SJML v2.0. The swimlane approach was adapted to present several actors. Therefore, the way of presenting actors has been updated. One can see each actor's journey by following each swimlane. An axis with time information for each touchpoint has been added to show the time flow of the service journey.

- **Swimlanes.** The swimlane concept was adapted to describe the service journeys of several actors. Touchpoints, actions, and system activities for an actor are allocated inside a horizontal grey bar. Several bars are collocated vertically, and the interactions are described with arrows and the colours and descriptions of the touchpoints.

- **Actor.** The boundary between customer and service provider has become ambiguous due to the involvement of many actors. All of them are actually customers of the EHR system. A symbol on the left side of a swimlane bar represents an actor. The text below the symbol specifies the role of the actor in the service journey.
- **Actor's journey.** An actor's journey can be seen by following a swimlane bar.
- **Time flow.** An axis with time stamps under the swimlane diagram represents the time flow and the occurrence dates of the touchpoints.

G. Preliminary Experience

We verified the visualisation of the referral process using SJML v2.0 by communicating with the chief nurse via emails and telephone conversation. We then presented the visualisation to a small group of experts (five employees) who work at the eHealth system producing company that produces the EHR system through a remote workshop using Lync. After the presentation, a set of questions was sent to the participants via email to collect preliminary feedback. The questions investigated whether the referral process and visual elements described using SJML v2.0 were understandable and whether the language would be helpful for creating a common understanding of the business process or workflow of the referral process. We received answers from two respondents. Thus, we have only anecdotal evidence of evaluation of the language based on this preliminary experience. The preliminary feedback from the participants regarding SJML v2.0 was analysed and summarised using thematic coding [40]. The summary is presented below.

Description of the process and visual elements using SJML v2.0:

- The language itself is relatively easy to understand
- The swimlanes to distinguish actors are simple, straightforward and easy to understand
- The symbols for actors and channels are easily understandable
- The use of different colours to distinguish the initiator and receiver is helpful for identifying the initiator

Creating a common understanding of the process and workflow using SJMLv2.0:

- The representations of SJML v.2.0 are suitable for discussing and establish a common understanding of the workflow and relationships among actors
- It would be better if the tool could also illustrate alternatives in a process.

V. DISCUSSION

Through our analysis of the related work, we found some weaknesses in the existing process modelling languages and the service design technique describing service delivery processes. First, there is a need to focus on the customer's perspective [1]. Second, there is a need to support the satisfactory description of the actual execution of the service delivery process [5][6]. Third, there is a need to sufficiently illustrate networked relationships in the service delivery process [7]. Fourth, there is a need for better visual expressiveness to describe the service delivery process explicitly [5][6][37]. Thus, there is a need for a new language or technique.

Relevant knowledge in the areas of service design, information visualisation, communication theory, and modelling languages have formed a theoretical basis for our approach to find a solution for these needs. Based on this approach, we developed a service domain specific process modelling language (Service Journey Modelling Language, SJML) to cover the aforementioned gaps in BPM for services. The language was developed and evaluated through case studies with partner companies, following the three phases of design science (problem analysis, solution design, and evaluation) in an iterative manner. For evaluations, we used each version of our language to describe existing or new services.

We evaluated the first version of our language (SJML v1.0) through a workshop with library staff by describing simple service journeys (book-loan services in a library) [1]. During the workshop we found that the participants did not have any problems in understanding SJML v1.0, and they could construct service journeys with our language. Feedback from the participants indicating the weakness of SJML v1.0 (a need for decision points) was addressed when we developed the next version of our language (SJML v1.1). SJML v1.1 was evaluated by describing more complicated service journeys from our research partner companies [8].

SJML v1.1 was updated by describing an even more complicated service journey (a referral process), which presents many actors and their interactions. The updated version of our language (SJML v2.0) adopted a sender-recipient concept and a swimlane approach. We visualised a patient history that was extracted from an EHR system using SJML v2.0 and presented the result to a small group of people via a remote workshop. We received feedback from the participants that the visual elements and the language itself were easy to understand and suitable for creating a common understanding of the workflow and relationships among actors involved in the business process.

To represent the customer's perspective, our language was developed based on a customer journey mapping approach that introduced the touchpoint concept. SJML v1.0 supported a sequential view to describe the expected journey. To support the description of the actual journey,

SJML v1.1 added a deviation view. SJML v2.0 added a new diagram type (swimlane view) to support the illustration of the networked relationships of several actors in a service delivery process. Each version of SJML expanded the graphical elements to better support the visual expressiveness.

VI. LIMITATIONS

The library service and referral process can vary in practice, depending on the organisation. In this paper, we conducted our case studies on book loan services in a Norwegian university library and a referral process in a Norwegian hospital; thus, SJML might not meet all the challenges necessary to describe similar services in other organisations.

For the case study on the referral process, we collected patient data based on existing patient care histories in an EHR system. Therefore, there might be a limitation in regards to covering all the touchpoints or actions that were not registered in the EHR system.

The initial feedback on SJML v1.0 was gathered through observations and a group discussion. Audio recordings were not used, and thus the discussion was not transcribed. Observers took notes when the group members talked, and the results were based on those field notes. Therefore, the descriptive and interpretive validity might be limited.

VII. CONCLUSION AND FUTURE WORK

Services usually have complex structures with several actors whose interests are intertwined. Aligning business processes with the customer's perspective is important for BPM, especially in the service field. To do this, a good understanding of interactions among the actors involved in service provision and consumption is needed. Several techniques for managing business processes to support the service delivery process were suggested in the context of process modelling and service design. However, there was a lack of support for describing the customer's point of view, the actual execution of service and the networked perspective of the service delivery process. In addition, there is a need to improve the visual expressiveness of the existing techniques.

Several versions of SJML were developed and evaluated in an iterative manner to strengthen the customer orientation, to facilitate collaboration among all involved actors and to support specification, description and analyses of service delivery processes. The requirements gathered from the evaluations provided useful input for further development of our language.

We believe that SJML v1.0 supplements BPM for services by aligning business processes with the customer's perspective [1]. SJML v2.0 supports the description of the actual journeys of actors and their networked interactions. Therefore, we expect that SJML will support BPM of complex services. Furthermore, the results from our preliminary experience support the contention that the visual representations of our language could be effective in creating

a common understanding of workflows and relationships among various actors in the business processes of services.

We intend to investigate the usefulness of SJML in BPM in future studies. Experiments or another case study with a larger number of participants would be used in further research.

ACKNOWLEDGMENT

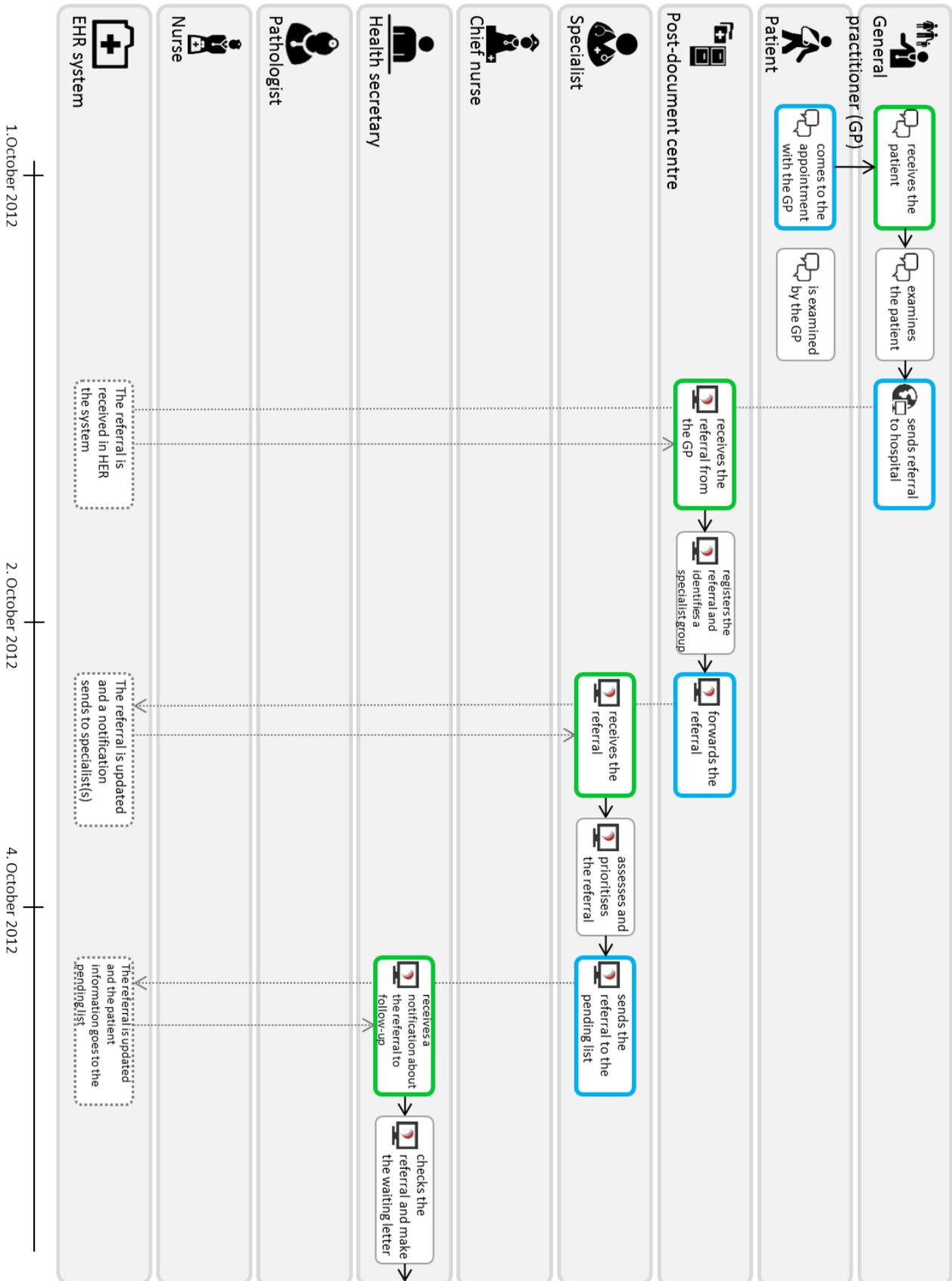
The research presented here has been conducted within the VISUAL project (2012–2016, project number 219606) funded by the Research Council of Norway and industrial partner companies, and involved SINTEF ICT and the University of Linköping. Thanks to Ida Maria Haugstveit for her significant contribution in developing the language and creating the visual diagrams for the case study, to Alma Culén and Andrea Gasparini for organising the library workshop, to our industrial partners for their contribution, and to the participants of all the workshops. We thank the reviewers for their useful suggestions and comments.

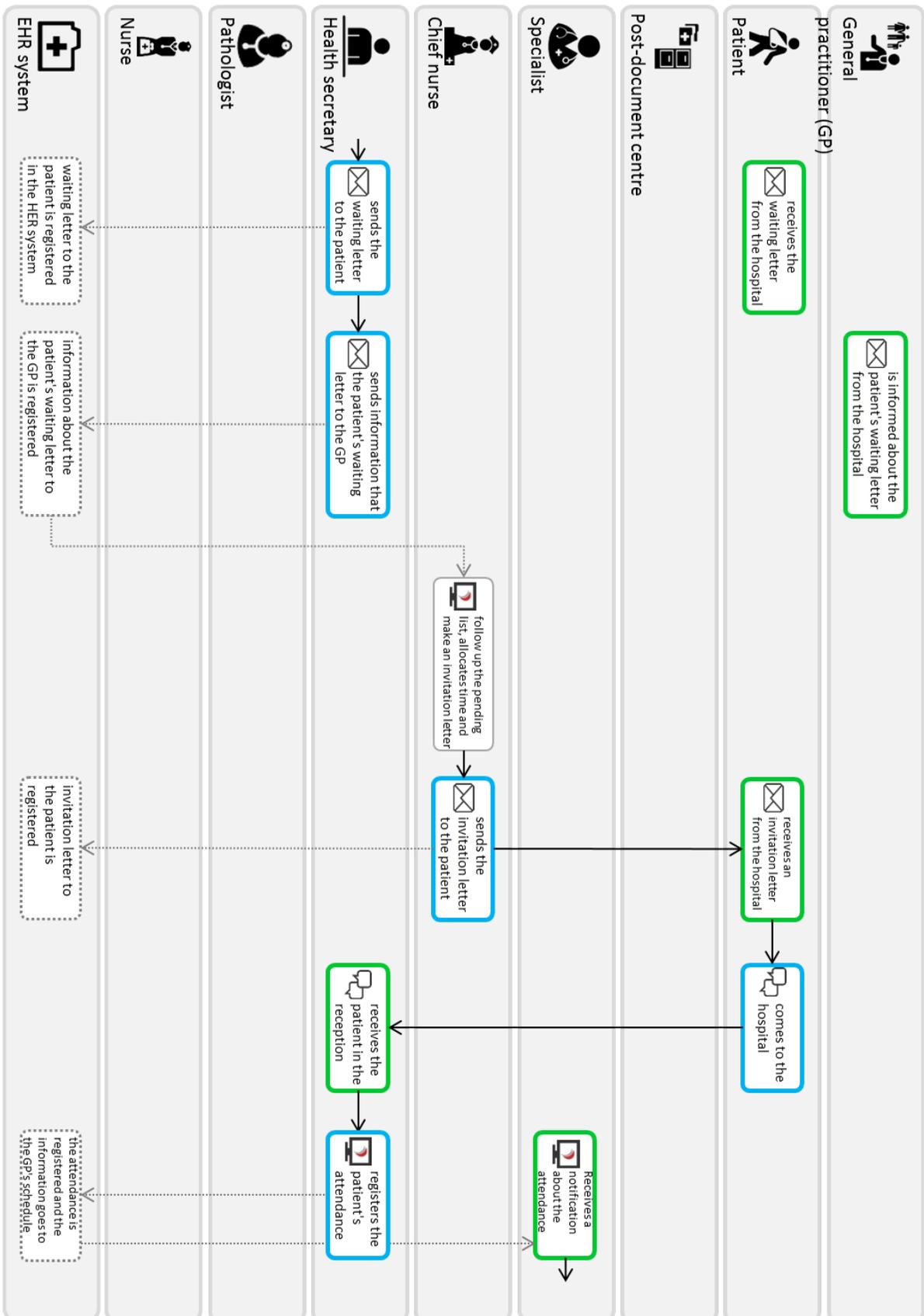
REFERENCES

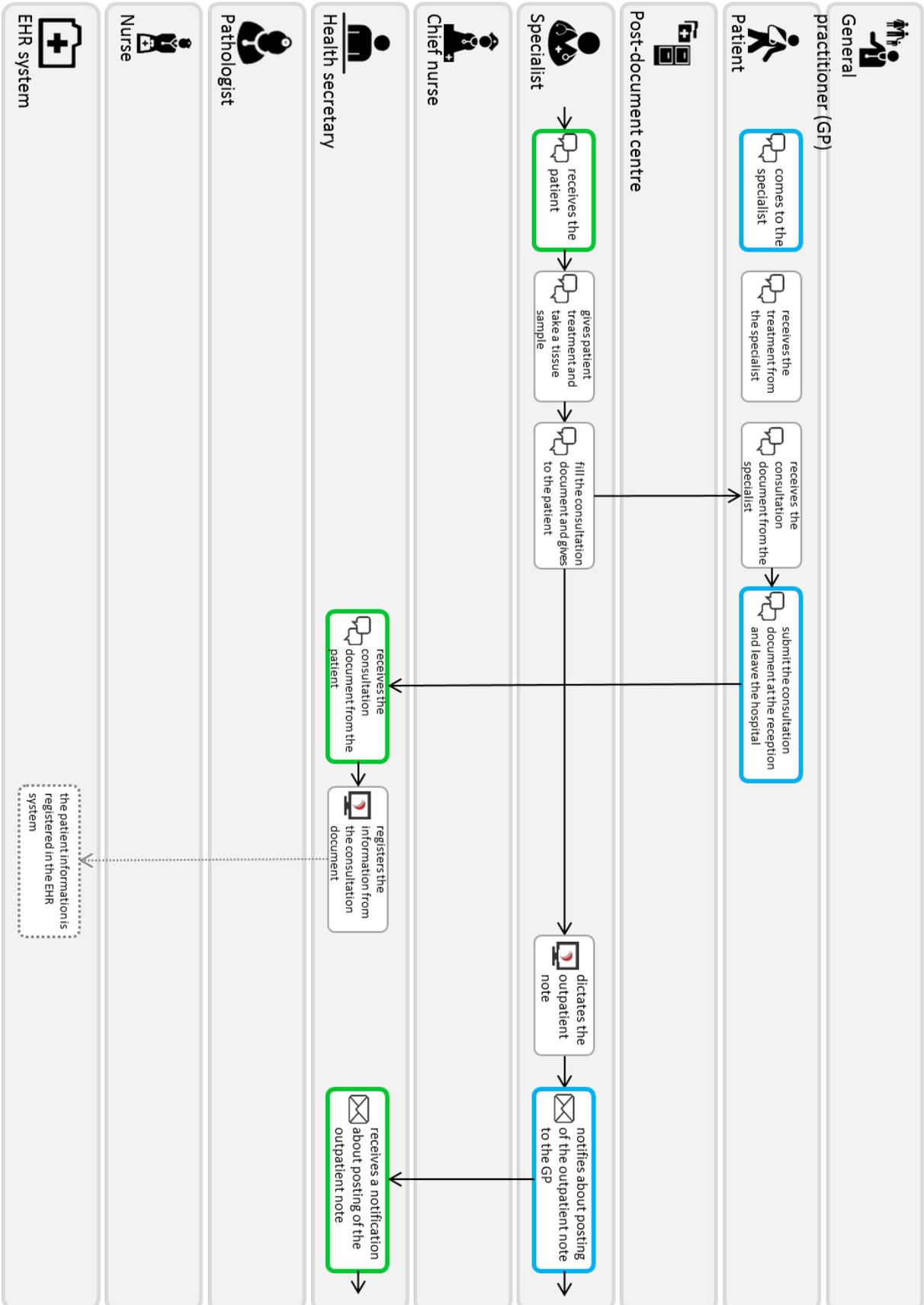
- [1] E. Lee and A. Karahasanovic, "Can Business Process Management Benefit from Service Journey Modelling Language?," The Eighth International Conference on Software Engineering Advances (ICSEA 2013) IARIA, Venice, Italy, Oct. 2013, pp. 579–581, ISSN: 2308-4235, ISBN: 978-1-61208-304-9.
- [2] T. P. Soubbotina and K. A. Sheram, Beyond economic growth: Meeting the challenges of global development. Washington, D.C., U.S.A.: World Bank Publications, 2000.
- [3] The statistical office of the European Union. *Almost 70% of employed persons in the EU27 worked in the service sector in 2011 - Product - Eurostat*. [Online]. Available from: <http://ec.europa.eu/eurostat/en/web/products-press-releases/-/3-05102012-AP>, [Accessed: 2015.12.01].
- [4] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, and M. Weske, "Business process management: A survey," in Business process management, 2678 vols., Wil M. P. Van der Aalst and Mathias Weske., Eds. Eindhoven, Netherlands: Springer, pp. 1–12, 2003.
- [5] L. Verner, "BPM: The Promise and the Challenge," Queue, vol. 2, no. 1, pp. 82–91, Mar. 2004, doi: 10.1145/984458.984503.
- [6] T. Wreiner, I. Maartensson, O. Arnell, N. Gonzalez, S. Holmlid, and F. Segelström, "Exploring Service Blueprints for Multiple Actors: A Case Study of Car Parking Services," The First Nordic Conference on Service Design and Service Innovation, Oslo, Norway, Nov. 2009, pp. 213–223, ISSN: 1650-3740, ISBN: 978-91-7519-771-5.
- [7] S. S. Tax, D. McCutcheon, and I. F. Wilkinson, "The Service Delivery Network (SDN) A Customer-Centric Perspective of the Customer Journey," Journal of Service Research, vol. 16, no. 4, pp. 454–470, 2013, doi:10.1177/1094670513481108.
- [8] R. Halvorsrud, E. Lee, I. M. Haugstveit, and A. Følstad, "Components of a Visual Language for Service Design," The Fourth Service Design and Innovation Conference (ServDes. 2014), Lancaster, UK, Apr. 2014, pp. 291–300, ISSN: 1650-3740, ISBN: 978-91-7519-280-2.
- [9] WHO, "Referral Systems - a summary of key processes to guide health services managers." The World Health Organisation, 2013.
- [10] L. J. Menor, M. V. Tatikonda, and S. E. Sampson, "New service development: areas for exploitation and exploration," Journal of Operations Management, vol. 20, no. 2, pp. 135–157, Apr. 2002, doi:10.1016/S0272-6963(01)00091-2.
- [11] R. M. Saco and A. P. Goncalves, "Service Design: An Appraisal," Design Management Review, vol. 19, no. 1, pp. 10–19, Jan. 2008, doi: 10.1111/j.1948-7169.2008.tb00101.x.
- [12] H. Dubberly and S. Evenson, "Designing for service: Creating an experience advantage," in Introduction to Service Engineering, G. Salvendy and W. Karwowski, Eds. U.S.A: John Wiley & Sons, pp. 403–413, 2010.
- [13] M. Stickdorn and J. Schneider, This is service design thinking. Wiley, 2010.
- [14] M. J. Bitner, A. L. Ostrom, and F. N. Morgan, "Service blueprinting: a practical technique for service innovation," California Management Review, vol. 50, no. 3, pp. 66–95, 2008, doi:10.2307/41166446.
- [15] Oxford, The New Shorter Oxford English Dictionary. New York: Oxford University Press Inc, 1993.
- [16] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Readings in information visualization: using vision to think. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [17] Y. Rogers, H. Sharp, and J. Preece, Interaction Design: Beyond Human - Computer Interaction. John Wiley & Sons, 2011.
- [18] C. Chen and Y. Yu, "Empirical studies of information visualization: a meta-analysis," International Journal of Human-Computer Studies, vol. 53, no. 5, pp. 851–866, Nov. 2000, doi:10.1006/ijhc.2000.0422.
- [19] A. Karahasanovic, Supporting application consistency in evolving object-oriented systems by impact analysis and visualisation, Oslo, Norway: University of Oslo, 2002.
- [20] C. Ware, Information visualization: perception for design, 3rd ed. Morgan Kaufmann, 2000.
- [21] M. J. Bitner, V. A. Zeithaml, and D. D. Gremler, "Technology's impact on the gaps model of service quality," in Handbook of Service Science, P. P. Maglio, C. A. Kieliszewski, and J. C. Spohrer, Eds. Springer, pp. 197–218, 2010.
- [22] C. E. Shannon and W. Weaver, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, no. 1, pp. 379–423; pp. 623–656, Oct. 1948, doi:10.1145/584091.584093.
- [23] R. T. Craig, "Communication theory as a field," Communication theory, vol. 9, no. 2, pp. 119–161, 1999, doi:10.1111/j.1468-2885.1999.tb00355.x.
- [24] C. S. Souza, The semiotic engineering of human-computer interaction. The MIT press, 2005.
- [25] R. Jakobson, "Linguistics and poetics," in Style in language, T.A. Sebeok, Eds. Cambridge, MA, U.S.A.: Massachusetts Institute of Technology, pp. 350–377, 1960.
- [26] D. Harel and B. Rumpe, "Meaningful modeling: What's the semantics of 'Semantics'?", The IEEE Computer Society,

- vol. 37, no. 10, pp. 64–72, Oct. 2004, doi:10.1109/MC.2004.172.
- [27] M. Mernik, J. Heering, and A. M. Sloane, “When and how to develop domain-specific languages,” *ACM computing surveys*, vol. 37, no. 4, pp. 316–344, 2005, doi:10.1145/1118890.1118892.
- [28] K. Garland, *Mr Beck’s Underground Map*. Capital Transport, 1994.
- [29] IEEE, *Software and Systems Engineering Vocabulary*. [Online]. Available from: http://pascal.computer.org/sev_display/search.action [Accessed: 2015.12.01].
- [30] OMG, *UML 2.5 - Review*. [Online]. Available from: <http://www.omg.org/spec/UML/2.5/>, [Accessed: 2015.12.01].
- [31] H.-E. Eriksson and M. Penker, *Business modeling with UML: Business Patterns at Work*, vol. 1. New York, NY, USA: John Wiley & Sons, Inc., 1998.
- [32] OMG, “Business Process Model and Notation (BPMN).” The Object Management Group, Inc., 2011.
- [33] OMG, *Business Process Management Initiative*. [Online]. Available from: <http://www.omg.org/bpmn/index.htm>, [Accessed: 2015.12.01].
- [34] G. L. Shostack, “Designing services that deliver,” *Harvard business review*, vol. 62, no. 1, pp. 133–139, 1984.
- [35] G. T. Jun, J. Ward, Z. Morris, and J. Clarkson, “Health care process modelling: which method when?” *International Journal for Quality in Health Care*, vol. 21, no. 3, pp. 214–224, 2009, doi: <http://dx.doi.org/10.1093/intqhc/mzp016>.
- [36] J. C. Recker, M. Indulska, M. Rosemann, and P. Green, “How good is BPMN really? Insights from theory and practice,” *The Fourteenth European Conference on Information Systems (ECIS 2006)*, Göteborg, Sweden, Jun. 2006, vol. 1, pp. 1–12.
- [37] A. Afrasiabi Rad, M. Benyoucef, and C. E. Kuziemsy, “An evaluation framework for business process modeling languages in healthcare,” *Journal of theoretical and applied electronic commerce research*, vol. 4, no. 2, pp. 1–19, 2009.
- [38] A. Polaine, L. Løvlie, and B. Reason, *Service Design: From Insight to Implementation*. Rosenfeld Media, 2013.
- [39] F. Segelström, *Stakeholder Engagement for Service Design: How service designers identify and communicate insights*. Linköping University Electronic Press, 2013.
- [40] R. Madden, *Being ethnographic: A guide to the theory and practice of ethnography*. Sage Publications, 2010.
- [41] D. F. Sittig, T. K. Gandhi, M. Franklin, M. Turetsky, A. J. Sussman, D. G. Fairchild, D. W. Bates, A. L. Komaroff, and J. M. Teich, “A computer-based outpatient clinical referral system,” *International Journal of Medical Informatics*, vol. 55, no. 2, pp. 149–158, Aug. 1999, doi: [http://dx.doi.org/10.1016/S1386-5056\(99\)00027-1](http://dx.doi.org/10.1016/S1386-5056(99)00027-1).

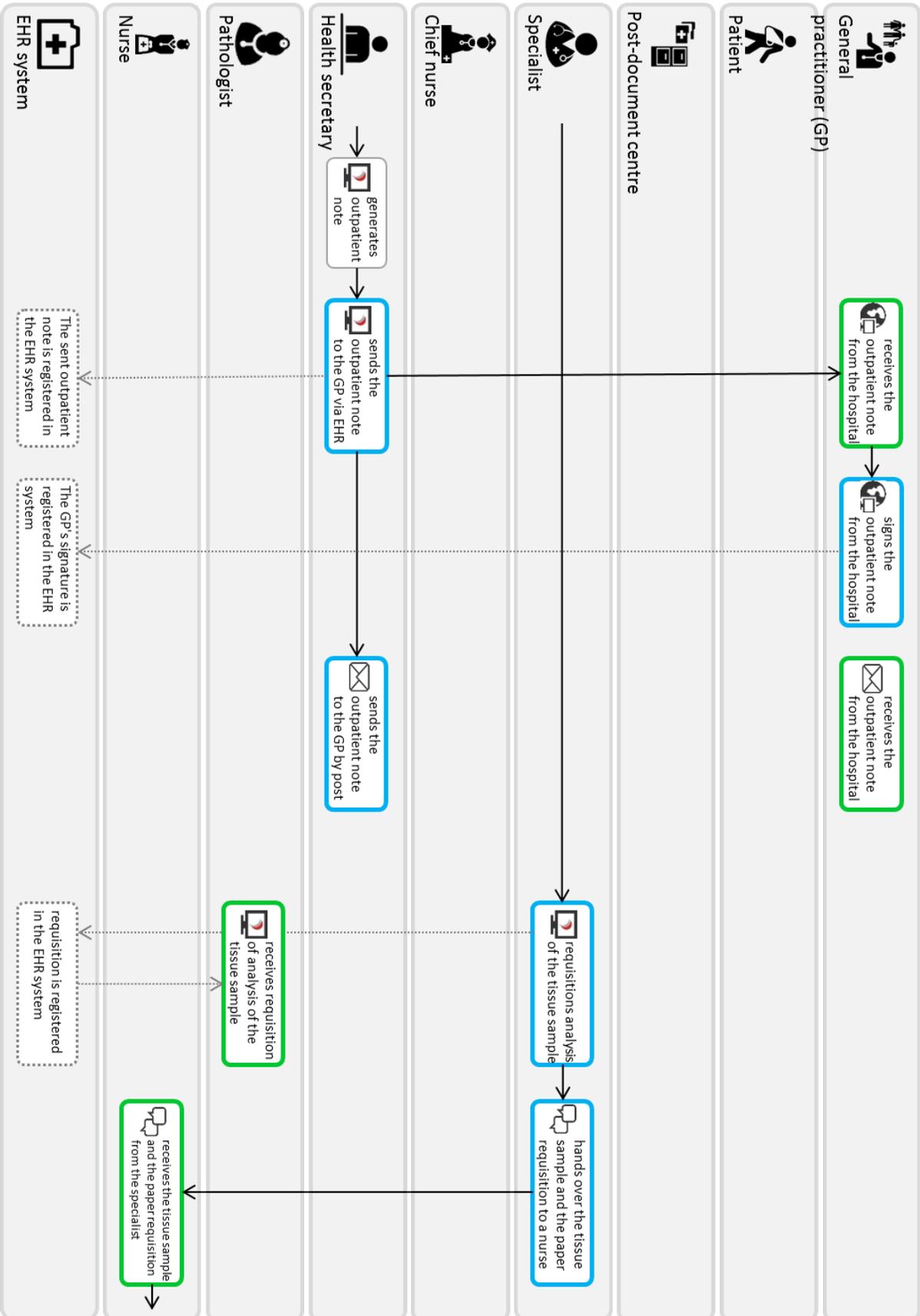
Appendix A.



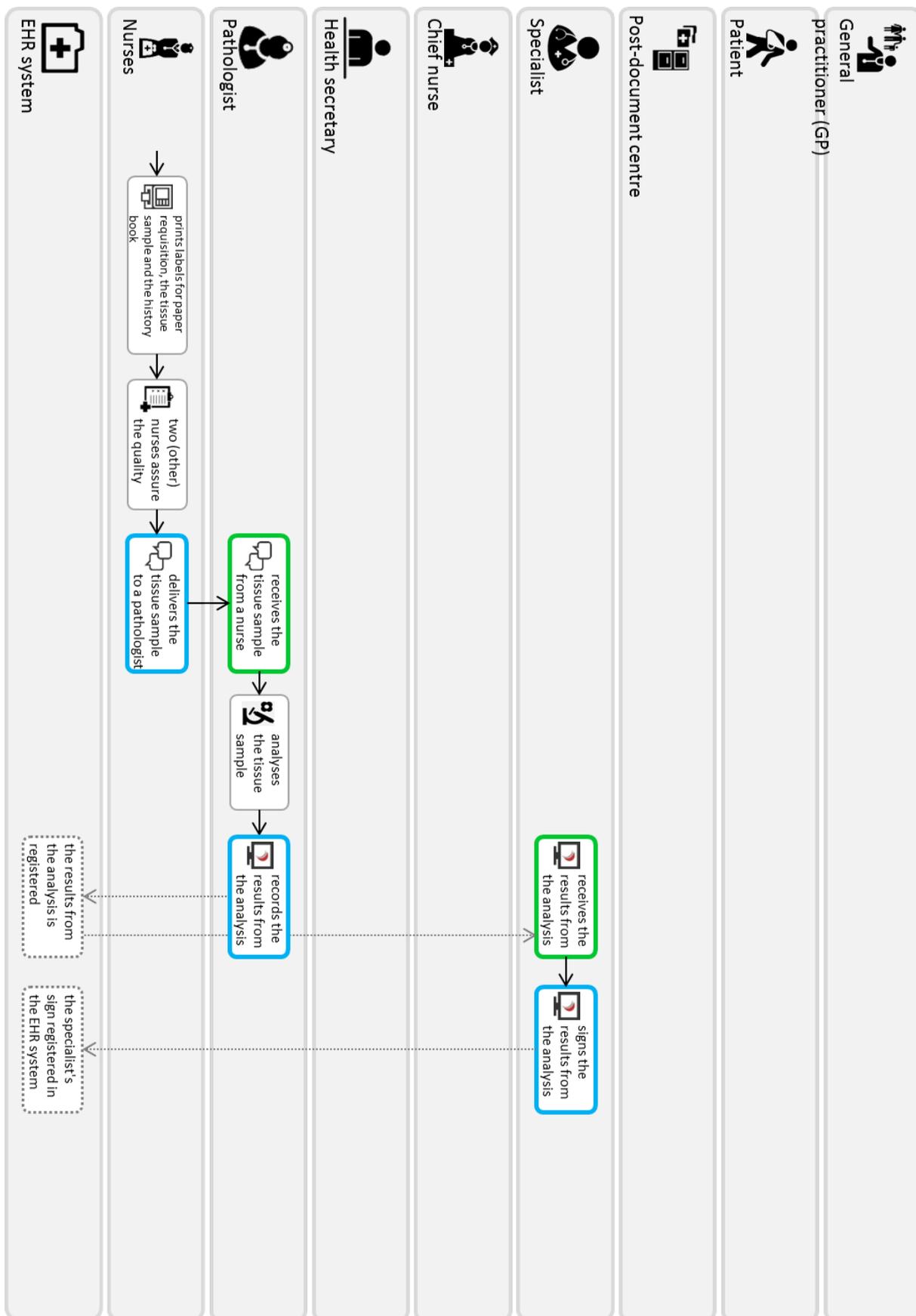




(6:September 2013)



(6:September 2013)



Reconfigurable Equiplets Operating System

A Hybrid Architecture to Combine Flexibility and Performance for Manufacturing

Daniël Telgen*, Erik Puik*, Leo van Moergestel*, Tommas Bakker*, John-Jules Meyer†

*Research Centre Technology & Innovation
HU University of Applied Sciences Utrecht
Nijenoord 1, Utrecht, The Netherlands
Email: daniel.telgen@hu.nl

†Dept. of Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands
Email: J.J.C.Meyer@uu.nl

Abstract—The growing importance and impact of new technologies are changing many industries. This effect is especially noticeable in the manufacturing industry. This paper explores a practical implementation of a hybrid architecture for the newest generation of manufacturing systems. The paper starts with a proposition that envisions reconfigurable systems that work together autonomously to create Manufacturing as a Service (MaaS). It introduces a number of problems in this area and shows the requirements for an architecture that can be the main research platform to solve a number of these problems, including the need for safe and flexible system behaviour and the ability to reconfigure with limited interference to other systems within the manufacturing environment. The paper highlights the infrastructure and architecture itself that can support the requirements to solve the mentioned problems in the future. A concept system named Grid Manufacturing is then introduced that shows both the hardware and software systems to handle the challenges. The paper then moves towards the design of the architecture and introduces all systems involved, including the specific hardware platforms that will be controlled by the software platform called REXOS (Reconfigurable Equiplets Operating System). The design choices are provided that show why it has become a hybrid platform that uses Java Agent Development Framework (JADE) and Robot Operating System (ROS). Finally, to validate REXOS, the performance is measured and discussed, which shows that REXOS can be used as a practical basis for more specific research for robust autonomous reconfigurable systems and application in industry 4.0. This paper shows practical examples of how to successfully combine several technologies that are meant to lead to a faster adoption and a better business case for autonomous and reconfigurable systems in industry.

Index Terms— *Flexible Manufacturing Systems; Multi-agent systems; Autonomous agents; Reconfigurable architectures.*

I. INTRODUCTION

Computers are continuously changing industry, and whereas in the past this has created opportunities for improved logistics and overview like SCADA (Supervisory Control and Data Acquisition) it is now starting to change the industry itself. This paper is based on our original work [1], which was presented at the INTELLI conference. Computers are not only supporting existing mechatronical systems, but are fundamentally chang-

ing the processes in how they are used. This is the basis for many changes in the field of manufacturing that are known under a variety of names, including 'Smart Industry', 'Agile Manufacturing', 'Industry 4.0', and Cyberphysical systems.

With the use of more computing power in manufacturing systems it is easier to integrate "intelligent", i.e., dynamic behaviour by using microsystems, i.e., sensors and actuators, together with advanced software to interpret the sensed data and act accordingly. This creates a robotic system that can dynamically interact and act with its environment. However, the dynamic behaviour of this systems can also increase complexity [2]. Therefore, it is important to create a balanced architecture that, on the one hand, has a high performance to control the hardware, i.e., dynamically interpret and interact with its environment in real-time and, on the other hand, is not so complex that it will be difficult to use and shows unexpected, i.e., unsafe or unwanted, behaviour.

From a hardware perspective there are also several changes that are occurring, as the costs for developing such complex systems have to be earned back and, therefore, it is necessary to reuse systems as often as possible. This has led to the creation of modular systems, where the modules can be used as building blocks that can be combined for a specific purpose. If the modular systems are standardized and well-documented they can also be easily reconfigured to provide a range of options. Since a well-defined module could be seen as a black box, it also lowers complexity of the overall system, since functionality can be abstracted on a higher level [2].

The structure of the paper is as follows: The next section will give further information for the motivation and how the research was conducted. Section III will provide more insight overview of current technologies and paradigms that are involved when researching smart technologies for manufacturing. After this overview the problem will be investigated more closely in Section IV, which will result in a number of specific research questions. Section V will then continue with the chosen approach and provide the basis for the requirements in Section VI. The requirements will then lead as input for an overview of several technologies and concepts in

Section VII. Section VIII will then describe the design choices that have been made. The design triggered the development of a software architecture and hardware platform has been implemented and described in Section IX. Finally, the platform will be evaluated with several performance results in Section X. These results and critical points of the platform will be discussed and concluded in the final Section XI.

II. RESEARCH MOTIVATION

Due to the obvious advantages of using smart systems, many companies and research groups are experimenting and conducting Research and Development with several projects. However, the success in industry itself has so far been limited [3]. This is due to a number of reasons, including the complexity of such systems with the high initial investment costs, the expertise needed to create such systems, and the difficulties to create a business case for dynamic systems. Schild and Bussmann show this in a case where a successful self-organized manufacturing system was setup at Daimler-Chrysler. They state that while the system was successfully implemented it was discontinued because a technical advantage is not always a measurable economical advantage [4]. When continuing this research in this field, therefore, it is important to take this aspect into account. In this research, this is done by taking into account several techniques and requirements that will lower the hardware costs, and also by targeting the manufacturing means at an even wider variety.

Besides industrial parts there is also a change in retail industry. Mass customization is slowly beginning to become a standard. Wind even introduces the concept of 'customerization'. This is a new business strategy that combines personal marketing strategies, with mass customization. Figure 1 shows that customerization comes through a combination of standardization, personalization and mass customization. Wind mentions that for successful customerization requires the integration of multiple processes, including operations and R&D [5]. He also states that increasing the digital content of everything the company does is one of two critical aspects that should be considered.

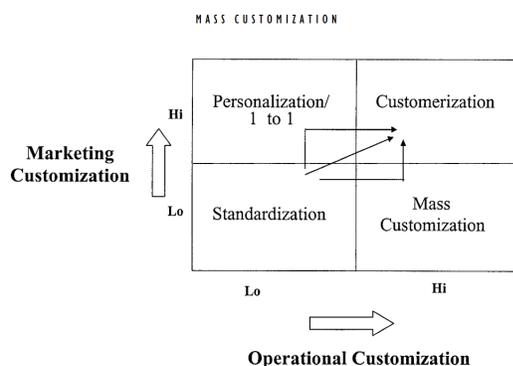


Figure 1. Combining personalisation and mass customization towards a new business strategy [5].

This paper will take a step further into the field of flexible

reconfigurable manufacturing by laying the foundation for a flexible and reconfigurable architecture for high-mix, low-volume manufacturing that could enable customerization.

A. Research Approach

As observed in the introduction, technological advances and changes in industry are starting to have an increasing impact on the manufacturing industry. However, a true paradigm shift has not yet occurred, likely because the maturity and efficiency of these new technologies has yet to be proven. As Leitão states in the conclusion of his survey: 'The challenge is thus to develop innovative, agile and reconfigurable architectures for distributed manufacturing control systems, using emergent paradigms and technologies that can provide the answer to those requirements' [3]. Leitão also identifies some specific issues in this field, including:

- 1) The need for mature and proven technology - the majority uses laboratorial control applications without the need of physical devices [6].
- 2) Reconfigurability mechanisms - what architecture will support the society of distributed entities? [3].
- 3) Development-related aspects - current platforms have limited scalability and robustness [7].
- 4) Prediction in disturbance handling systems - the integration of prediction mechanisms with identification and recovery of disturbances that can prevent these problems [3].

The research in this paper responds to these factors by including the hardware systems and the practical implementation to show a feasible system that can be used for industry. Together with the hardware, several emulators and simulators will also be developed to test various aspects of the proposed system. Besides using the simulation for benchmarking, it will also be used to predict problems like collisions in the changing and possibly chaotic environment.

B. Hypothesis

The hypothesis is that new (software) technologies could be used to create a new flexible manufacturing paradigm that is (cost-) efficient and stable. The flexibility will have to provide for a much shorter time to market and an increased variety in products. The challenge for this hypothesis is to keep the complexity (and, therefore, the practical applicability) of the smart and flexible approach under control. This has to be proven by developing a proof of concept that shows the abilities, performance and stability of such an 'agile' architecture. To prove the feasibility and practical implementation the system will be fully built including low-cost hardware designed for this purpose.

C. Research Methodology

The study will be largely based on applied research. An architecture and several hardware and software systems will be developed to be used as a proof of concept. This system will be the basis of future research and will be combined with

several emulators to be tested in a large variety of cases. The study will combine quantitative and qualitative elements:

- performance: quantitative research based on empirical data using experiments from live proof of concepts and simulators.
- abilities: qualitative research - comparing designs and architecture performance based on correlation.
- processes: qualitative comparison based on cases with (partly) quantitative data.

The research project will start with a 'top-down' approach, i.e., deductive research, that will test the proposition set in the next subsection. However, the platform will be created 'bottom-up' and as such is hoped to be a basis for inductive research to create new insights in the future for the use of reconfigurable systems and application of self-organizing manufacturing platforms.

D. Propositions

To focus the research, a concept has been created for a manufacturing platform that is based on several basic principles:

- 1) A range of products can, in principle, be dynamically built on demand - i.e., the machines offer a (wide) range of services where any product that can be made with these services can be manufactured.
- 2) Each system (with its own purpose) is autonomous - i.e., both products as manufacturing systems have no strong dependencies and can act autonomously.
- 3) Hardware should be reconfigurable. - i.e., both hardware and software modules within a system should be able to be changed with limited downtime. In the case of the software, compiling code should be required for a reconfigure action.
- 4) Machines should be low-cost and single-purpose. - i.e., the flexibility that is offered should be done with limited investment costs to guarantee experimental use and a valid business case.
- 5) System behaviour should be transparent and safe. - The flexibility and dynamic behaviour of the system must be guaranteed not to lead to a high risk of use.

The idea is to create a range of products that can be built on demand, i.e., all products can be manufactured ad hoc so long as the parts and required services to assemble them are available in the manufacturing systems or 'grid'.

The principles also focus on limiting complexity by creating a minimum amount of interdependence between systems. This results in a concept that has been called 'grid manufacturing', where each manufacturing system delivers a service to a product. Since products and the manufacturing systems have their own purpose, i.e., the machine delivers a service, the product wishes to be produced, they are both autonomous and will work together dynamically. Hence, the system will not be a 'production line', since the need for services will depend on the specific product demand.

Since the products will be manufactured dynamically without any specific programming it is important that they are

still produced safely and according to the specifications. The products will schedule themselves in negotiation with the manufacturing systems and, therefore, it is unknown which services are required. This makes it difficult to establish the demand, and therefore, the chances are that some manufacturing systems have a higher load than others.

III. GENERAL RESEARCH OVERVIEW

This section provides a theoretical framework and overview for the current research. It discusses a number of paradigms and technologies that could be applicable for the current study.

A. Manufacturing Paradigms

Many paradigms have emerged that have been of influence in the manufacturing industry, the most influential have been the three main paradigms, see Figure 2.

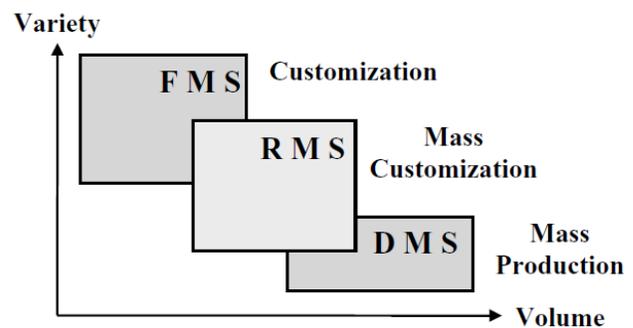


Figure 2. The three main manufacturing paradigms [8].

Dedicated Manufacturing Systems (DMS) are the classic way of mass-production. In this paradigm, all manufacturing systems are developed for a specific single-purpose goal with limited to no dynamic properties. This creates a cost-efficient system for producing high volumes of a single product over a longer timespan [9]. The requirements stay the same, therefore, DMS are known to have a high performance and limited initial costs.

Flexible Manufacturing Systems (FMS) offers dynamic behaviour, which it uses to react to changes. Usually, Flexible manufacturing systems offer a single purpose where the machine has the ability to perform one action. This ability is combined with sensors like a vision or dynamic routing system so it can adapt certain parameters, e.g., the position of a product. This creates more flexibility at the disadvantage of complexity and cost to the initial implementations.

Reconfigurable Manufacturing Systems (RMS) are unique in the perspective that the functionality of the system itself can be adapted [10]. As shown in Figure 2, they are positioned between FMS and DMS. However, since RMS provide a way to integrate change within a system it is expected that they will become more flexible over time [11]. ElMaraghy notes that the key characteristics of RMS include modularity, integrability, scalability, convertibility, and diagnosability [12]. These characteristics have to be taken

into account when defining the requirements for a flexible manufacturing architecture.

Besides the main three manufacturing paradigms, there are also many other paradigms and methods of interest.

Agile Manufacturing (AM) is characterised by the integration of customer and supplier for both product design, as manufacturing, marketing, and support services [13]. An agile manufacturing environment creates processes, tools, and a knowledge base to enable the organisation to respond quickly to the customer needs and market changes whilst still controlling costs and quality [14].

Manufacturing As a Service (MAAS) is a concept to deliver customizable and on-demand manufacturing. This is closely related to manufacturing clouds, where factories and their IT infrastructure are interconnected to create an infrastructure where ad-hoc products are being made [15].

Holonic Manufacturing is seen as an alternative to hierarchical management of manufacturing systems. It focuses on modularization and 'plug and play' capabilities when developing or using manufacturing systems [16]. Holonic manufacturing systems are often implemented using Multi Agent Systems [17], which will be discussed later in this paper.

Noteworthy are also a number of concepts that are closely related to this research:

Smart Industry and **Industry 4.0** are often used as a concept that combines industrial systems with properties like the 'internet of things' and other cloud related services. They depict the vision of smart factories that consist of **cyber-physical systems**. Cyber-physical systems (CPS) are collaborating computational (virtual) elements that control physical entities. i.e., basically a virtual entity with its virtual world image that uses its own world image to control and interpret the physical world and operate a physical counterpart in this environment. This virtual entity is commonly an embedded system within the system that it controls.

IV. PROBLEM DESCRIPTION

While many paradigms and technologies show promise they are not yet considered mainstream in industry. As mentioned before, the initial investment costs and complexity are factors for this problem. However, while business is important, this particular paper will first focus on the technological aspects that are required to create a basic software platform as a basis for further research and test cases for industry. For this platform a number of main challenges have been identified:

- 1) Architectural performance / intelligence gap - The platform should both be able to show 'intelligent' behaviour and have real-time performance to control the hardware.
- 2) Abstract Services - To use manufacturing as a service and limit complexity the hardware can not be 'known' by the product.
- 3) Reconfigurable systems - It should be possible to quickly adapt the hardware and reconfigure the system by changing its hardware modules.
- 4) System Behaviour - While systems should be autonomous and modular and work in a dynamic 'chaotic'

environment their behaviour should also be predictable and 'safe'.

The first problem will be the main focus of this paper. How can you create an architecture that combines the dynamic behaviour and flexibility for high-level functionality, i.e., understand its environment and cooperate with other systems in the grid, and low-level functionality, i.e., high performance hardware control and algorithms. These different functionalities are based on different behaviour and therefore have different requirements. High-level functionality is based on abstract cognitive processes that use networked data and slow heuristic processes. Low-level processes are based on strict rule based systems that have a direct impact on the actuators. As such they are usually written in native code using real-time systems. While native code could grant a higher performance it is also more difficult to develop. Additionally, it is important that the high-level functionality will have no performance impact on the low-level systems.

The next problems will be taken into account and seen as preconditions for the proposed architecture in this paper. Their impact on the architecture will be discussed. However, the solutions and research conducted to solve these specific problems are not within the scope of this paper itself and will be published in detail in future work.

The second problem focuses on the use of the manufacturing systems. Since the grid manufacturing concept asks for autonomous systems the product is not aware of which manufacturing system will produce it beforehand. As a result both product and the manufacturing system are not designed specifically for each other. Hence, to be able to use the service for a product they should be able to interface with and understand each other. This asks for an ontology that both product and manufacturing system can use. The architecture should take into account which services and limits it can provide and match these to the requirements of the product.

The third problem focuses on the reconfigurable aspect of the systems. Since demand can change it is important to adapt the systems to the (possibly new) demand. To create maximum flexibility the system should be easy to adapt and if possible automatically update its use and services so they can become unavailable to the products in the grid.

The last and fourth problem focuses on the system behaviour. Since products are unknown and manufacturing hardware can be reconfigured there is a large dynamic in a grid. Hence, it is difficult to define its exact behaviour. To be sure of the exact manufacturing specifications and safety aspects it is required to create specifications and procedures for action that a hardware module can perform. A system should be created that defines the behaviour and describes how it will act during diverse situations like starting up/shutting down or errors.

A. Research Questions

The research question will focus on the main problem of this paper, the creation and specification for a flexible software architecture that will provide both performance for low-level, and flexibility for high-level functionality.

- 1) What technologies are available for use in the proposed concept for grid manufacturing?
- 2) What requirements are necessary for a software architecture for smart industry?
- 3) How can low-level performance and high-level flexibility be combined?
- 4) Can such an architecture be scalable?
- 5) Is the proposed concept feasible for near future use in industry?

V. CONTEXT - THE CONCEPT

The main concept is based on the philosophy of cyber-physical systems. A grid will consist of three main types of systems: several logistic services (including autonomous transport systems), autonomous manufacturing systems, and lastly the products that will also be cyberphysical systems. Figure 5 shows the concept of a grid with autonomous (cyberphysical) systems. The reconfigurable manufacturing systems are called 'equiplets'.

Classic manufacturing is based on a Line Cell Module Device (LCMD) model as shown in Figure 3. The model represents a modular manufacturing process based on 4 hierarchical levels. The line is literally a 'manufacturing line' that is made up of a number of cells where a specific job is performed. A Cell commonly uses multiple modules which perform specific actions, e.g., pick & place. The module can be decomposed even further into devices, e.g., a pick & place module will likely consist of several sensors and actuators, each sensor and actuator can be seen as a device. The LCMD model is optimised for cost efficient manufacturing of products that are made in high quantities. Since this is a linear model where products are made in a line, any change at any level will influence the entire manufacturing process. Hence, the product as well as the manufactured process will have to be matured completely before actual mass manufacturing can start.

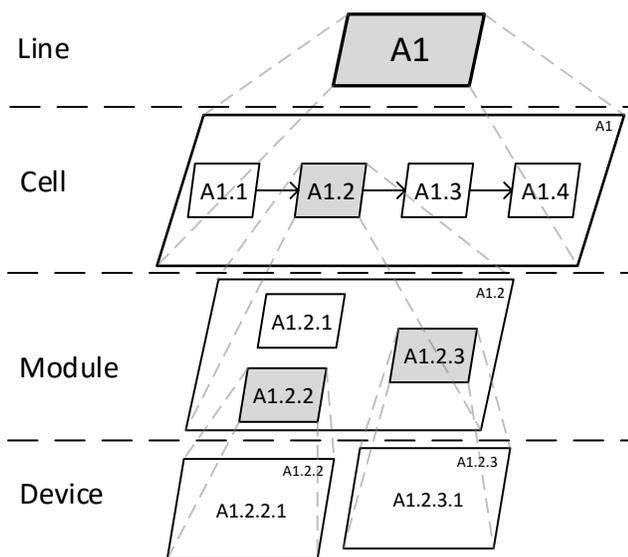


Figure 3. Classic Line Cell Module Device (LCMD) structure.

The concept of Grid manufacturing provides the opportunity to dynamically adapt both product and equipment at any level where the overall impact will be limited as much as possible. This is performed by autonomous reconfigurable systems that provide generic services that products can use. All systems in the grid should cooperate to become self-organizing. Because of the reconfigurable aspect of these systems they are named *equiplets*. Equiplets are not arranged in a line, but in a grid to emphasize that they can be used sequentially based on the current dynamic demand, i.e., dynamic in the sense that different products can be made at any time using equiplets in any order using (soft) real-time negotiation and scheduling to plan how a possibly unique product will be manufactured. Figure 4 shows a rendering of a grid with 12 equiplets, where every equiplet can have a different configuration to provide a variety of services that are required for the manufacturing process. Note that a grid does not require to have any specific form; depending on the demand they can be placed in any relative position based on the local logistic setup of the factory.

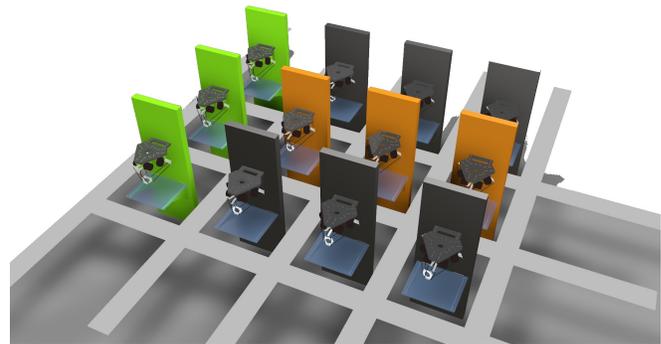


Figure 4. Example of a Grid Structure.

Equiplets provide capabilities that are based on the configuration of the specific modules that are installed. In this context, reconfiguration is defined as adding/removing/changing modules within the equiplet so as to change its capabilities. This includes both the physical change as the adaptation and configuration of the software to control the equiplet.

In contrast to LCMD, this architecture for Grid Manufacturing is called the Grid Equiplet Module (GEM) Architecture, as shown in Figure 6. With GEM the systems are loosely coupled, the Grid layer provides services to the autonomous equiplets. Modules are commonly designed as Components of the Shelf (COTS).

Besides delivering flexibility the concept also introduces a manner to bring the product designers and production experts closer together. In the past, lines were made specifically for one product and as such it would come at a high cost to take a working line off-line to create a prototype for a new product. Since grids can dynamically handle various products in parallel a product designer is able to use the same manufacturing equipment that is used for the final manufacturing to create prototypes and test the production phase. This shortens the time-to-market and lowers costs.

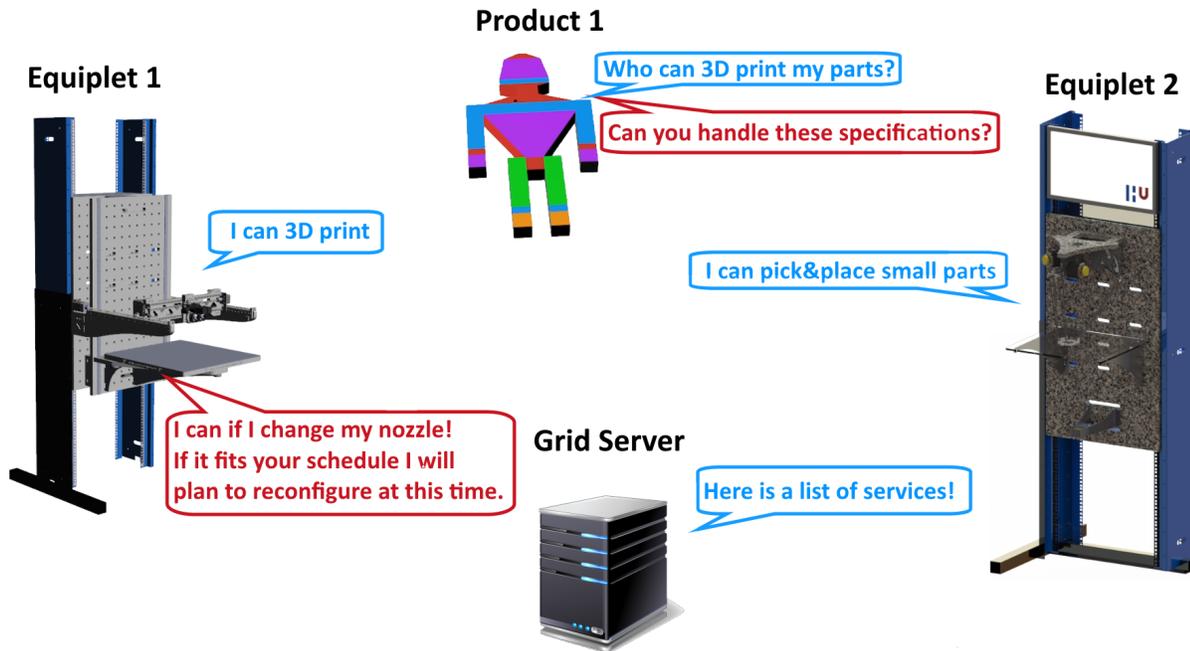


Figure 5. The simplified concept of grid manufacturing.

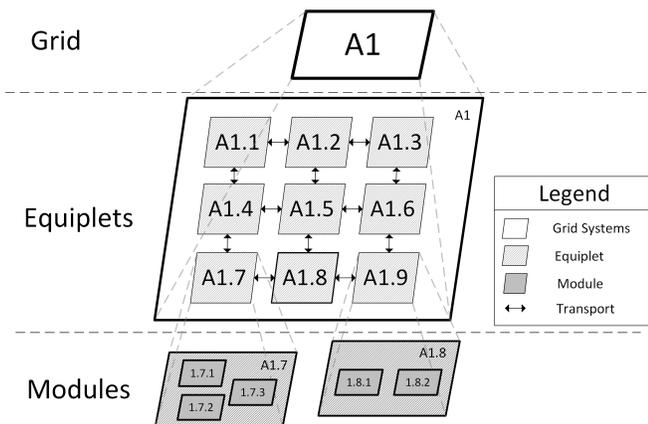


Figure 6. GEM Architecture.

To further work out the required platform and analyse the success of this concept the requirements should first be discussed.

VI. REQUIREMENTS

The requirements of grid manufacturing will be split into different parts. First, the preconditions and functional requirements of all levels within the GEM architecture. Then the products themselves, and finally the processes that are required to be active within the grid to fulfil its function.

Since the design of complex systems as used in grid manufacturing is challenging, the requirements are loosely based on the Axiomatic Design methodology developed by MIT [18]. Axiomatic design uses design principles or axioms, i.e.,

premises or starting points for reasoning. In Axiomatic design the characteristics needs are translated into four domains:

- Customer Domain - Customer Attributes (CA)
- Functional Domain - Functional Requirements (FR)
- Physical Domain - Design Parameters (DP)
- Process Domain - Process Variables (PV)

A. Functional Requirements

The functional requirements in Axiomatic Design are given by answering the question, 'what should the system do?'. This is placed in the scope of all software systems for multiple autonomous reconfigurable manufacturing machines. During the decomposition phase this clustered to a three level decomposition for the manufacturing system and a fourth for the product entity that represents the product:

- The Grid - A decentralised system where Equiplets and Products cooperate.
- Equiplet - An autonomous modular reconfigurable single-service low-cost manufacturing machine.
- Module - A hardware module that provides one specific function within an equiplet.
- Product - The cyberphysical entity that will represent the product.

B. Grid Level

The grid should be able to:

- GFR1 offer services to a variety of products.
- GFR2 validate and assess its own efficiency.
- GFR3 adapt (remove or add) services / equiplets with limited to no interference to other products.

- GFR4 provide for product transport dynamically between each equiplot.

C. Equiplot Level

An equiplot should be able to:

- EFR1 provide a specific service to a product.
- EFR2 be reconfigured (adding or removing of modules - to provide a different service).
- EFR3 work autonomous, i.e., it has no strict dependencies with other equiplots or create interferences for other equiplots.
- EFR4 automatically adapt its software when modules are added or removed.
 - EFR4A let its new service (capability) known to the grid.
 - EFR4B update its system behaviour and safety software.
- EFR5 translate abstract instruction from a product and translate it to instructions for its own specific hardware modules.
- EFR6 efficiently control the hardware in real-time.

D. Product representation entity

A product representation (in its manufacturing phase) should be able to:

- PFR1 coordinate its own production.
- PFR2 know of which parts it requires to be completed.
- PFR3 know which (abstract) services it requires to be assembled (production steps).
- PFR4 determine which services are available.
- PFR5 communicate with equiplots to determine if they can perform a production step.
- PFR6 create a (viable) schedule on how it will be produced.
- PFR5 log its production/assembly history.

E. Module Level

A module should be able to:

- MFR1 know its own characteristics.
- MFR2 accept and perform instructions from the equiplot.

VII. TECHNOLOGY COMPARISON

To create a platform for the proposed concepts let us first investigate some current technology:

Agent Technology The word Agent comes from the Latin word *agere*, which means: to act. Software agents, as shown in Figure 7 are entities that have their own interpretation of their environment on which they act autonomously. Hence, we uphold the definition of Wooldrige and Jennings: 'An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives'[19].

Therefore, some consider agents as 'objects with an attitude', since unlike an object an agent has control over its own behaviour. Agents can also be seen as a higher abstraction of objects, which makes them ideally suited to deal within

complex dynamic environments. Paolucci and Sacile noted that they can create a flexible, scalable and reliable production system [20].

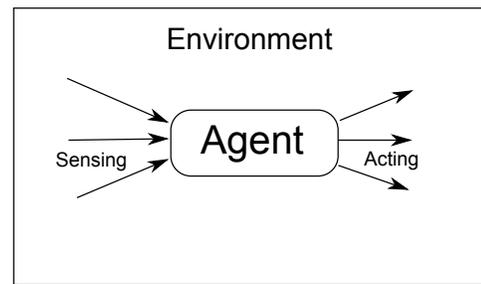


Figure 7. An autonomous agent in its environment.

Agents can be split in two main types:

- 1) Reactive agents
- 2) Reasoning agents

Figure 8 shows an example of a standard reactive agent cycle that perceives its environment through sensors, interprets it according to standard rules, and chooses an action accordingly and acts using actuators to change something in the environment.

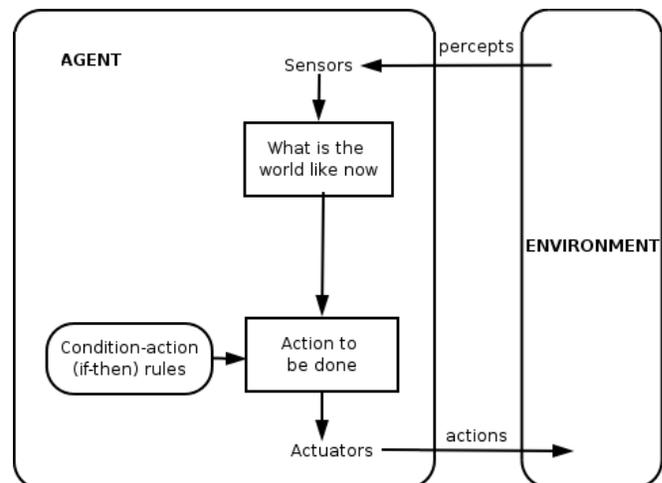


Figure 8. Simple reflex of an Intelligent Agent.

Reasoning agents exist in multiple types, the best known of which being the belief-desire-intention (BDI) agent, see Figure 9. The BDI agent uses the philosophy of Dennett and Bratman [21], [22]. The BDI agent uses its senses to build a set of beliefs, where its desires are a set of accomplishments that the agent wants to achieve. The BDI agent can choose desires that it wants to actively try to achieve, these are its goals. It then commits to a goal to make it into an intention, activating a plan that consist of actions that it will take to achieve its goal and thus satisfying its desire. Ideally, BDI agent uses the following sequence to achieve this [23]:

- 1) initialize-state
- 2) repeat
 - a) options: option-generator(event-queue)

- b) selected-options: deliberate(options)
 - c) update-intentions(selected-options)
 - d) execute()
 - e) get-new-external-events()
 - f) drop-unsuccessful-attitudes()
 - g) drop-impossible-attitudes()
- 3) end repeat.

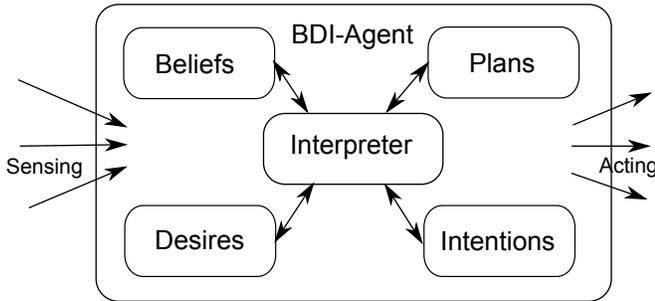


Figure 9. Beliefs Desire Intention Agent [24].

Figure 10 shows the concept of **Multi Agent Systems** (MAS), where multiple environments communicate and work within an environment. Agents interact and can cooperate or negotiate to achieve common goals. Either agent can have a specific role within a MAS and can interact with the other agents using specific permissions and responsibilities.

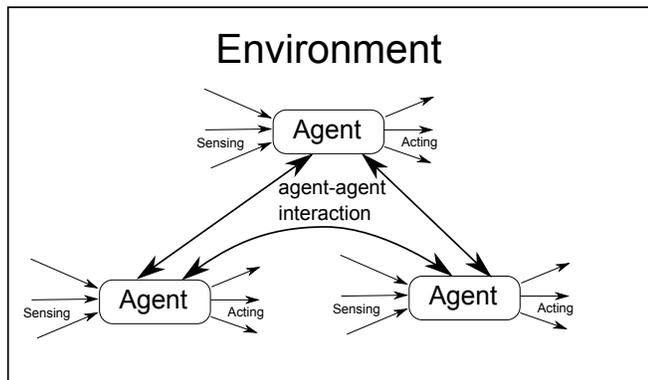


Figure 10. Multiple agents forming a Multi Agent System (MAS).

MAS can also be associated with **Environment Programming**. Environment Programming is seen as an abstraction where the environment is seen from the agent perspective. Objects in the environment that the agent interacts with are seen as programming models and are named 'artefacts'. This creates an extra abstraction where objects keep their abstraction layer and can be used effectively by the agents [25].

VIII. DESIGN

The platform that has been developed is called REXOS, which stands for Reconfigurable EQuipletS Operating System. For the design it was necessary to consider an important rule of Axiomatic Design:

TABLE I. Design Matrix that shows the relationship between FRs and DPs

	Hardware Platform	Intelligent Platform
High Performance	x	
Intelligent behaviour		x

- Axiom 1: The independence Axiom - Maintain the independence of the functional requirements
- Axiom 2: the information Axiom - Minimise the information content of the design

These axioms show why cooperating autonomous systems like MAS lower the complexity of a design, since many functional systems can be isolated in a single entity. However, this is not true for equiplets. Hence, they require specific attention in the design.

The domains are represented as vectors that are interrelated by design matrices.

$$\{FR\} = [A] \cdot \{DP\} \quad (1)$$

Where $\{FR\}$ is the vector of the Functional Requirements (What should it do), $\{DP\}$ the vector for the Design Parameters (What can satisfy the FR), and $[A]$ is the design matrix that hold the relationships between these two vectors:

$$\begin{bmatrix} FR1 \\ FR2 \end{bmatrix} = \begin{bmatrix} A11 & A12 \\ A21 & A22 \end{bmatrix} \times \begin{bmatrix} DP1 \\ DP2 \end{bmatrix} \quad (2)$$

As mentioned in the Problem Section, the architectural design has to be able to have high performance and be able to have intelligence behaviour. This is a common problem that is recognized in recent literature [26]. Based on the methodology of axiomatic design this urges us to think of how to decouple these properties.

Table I shows the relationship between the requirements and the solution. In this case, the requirements are decoupled through the creation of a *hybrid architecture* where multiple platforms are combined to potentially yield the best of two worlds [27]. This in contrast to a system where one platform is used where these requirements should be combined.

Hence, for the design, it is important to analyse a number of platforms and research how they can interface without becoming coupled. The next section will review a number of platforms and technologies that could become the basis of REXOS.

A. Choice of Technology

Grid Manufacturing can be seen as a complex system where many autonomous systems have to interact. This is one of the reasons why it is important to use autonomous entities to become as flexible as possible without creating too many interdependencies that increase the overall complexity of the system. As shown in Figure 11, a multi-agent system fits this requirement in that it offers a level of abstraction and limits the sphere of influence for an entity.

The use of a MAS seems a good option to choose as a basis for REXOS, if we investigate this further we can also look at the characteristics of a manufacturing environment [29]:

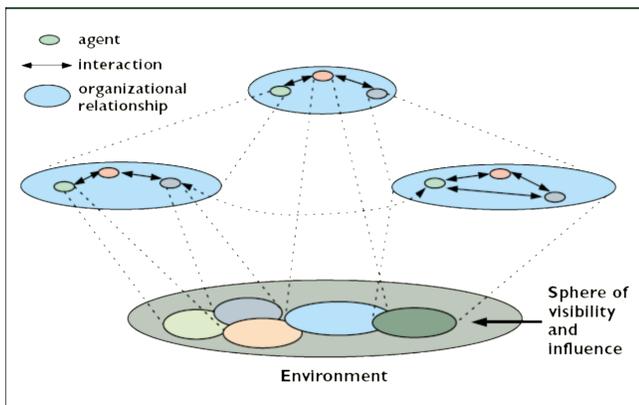


Figure 11. The sphere of influence within a MAS [28].

- 1) Autonomy
- 2) Cooperative
- 3) Communicating
- 4) Reactive
- 5) Pro-active

Together with the set requirements for equiplets and the grid, these fit perfectly into the concept of grid manufacturing. Besides the manufacturing processes themselves agents provide many more possibilities that are out of the scope for this paper, e.g., a product agent that stays with the product to analyse its behavior and offer problem solutions whenever possible [30]. The agent that can represent hardware could also be utilized to analyse efficiency and learn from the behaviour to optimize schedule times and other logistic matters.

B. Choice of platforms

Even though we choose to use MAS as a basis for REXOS, this does not fulfil all requirements that we require for grid manufacturing. MAS will provide a dynamic decision platform that will represent all systems. However, it is normally not suited for direct real-time control of hardware. Hence, it is important to investigate which platform could fulfil this requirement and to research how these platforms could be successfully combined. To approach this task let's first look at a number of agent platforms.

C. Agent platforms

The platform that is used for REXOS had to meet certain requirements as mentioned in the problem description. Several attributes also have to be satisfied, which are part of the Customer Domain:

- 1) The platform needs to be scalable.
- 2) For flexibility the platform needs to be able to change or add new agents during runtime.
- 3) The platform needs to be mature (for industrial application).
- 4) Performance needs to be sufficient to handle grid-wide logistics.
- 5) The platform should preferably be open source, but also applicable for industrial use with propriety sources.

Several agent platforms have been investigated:

- 2APL [31]
- JADE [32]
- Jadex [33]
- Madkit
- Jack
- Jason

The choice for the agent platform has become Java Agent Development Framework (JADE), since in JADE agents can migrate, terminate and start in runtime, also JADE has been widely adopted and has an active community. While JADE has no direct support for BDI agents it can be extended to add this when necessary in the future. Currently, the architecture does not force the use of the BDI. JADE is also compliant with the Interoperable intelligent multi-agent systems specifications standard FIPA. Which makes it possible to easily extend the MAS with other FIPA compliant systems Foundation for Intelligent Physical Agents.

D. Diverse platforms

Besides the agent platform, there is a need to combine it with other platforms to control the hardware and satisfy the Customer Attributes and Functional requirements.

Robot Operating System (ROS) is a software framework that provides services, tools and libraries for robots [34]. The framework has extensive support for a variety of sensors and actuators and offers hardware abstraction and low-level device control. ROS is free and open-source and uses nodes as software modules that communicate with messages. Nodes can be started and stopped in runtime, making it possible to adapt software modules at any time. ROS has been created to create general purpose robot software that is robust.

Robot Operating System (ROS 2.0) is currently under development and is being created to overcome some limitations of ROS 1.0, including real-time requirements and use for multiple robots.

MongoDB Due to the diversity and flexibility of the grid it is difficult to define all schemas that relational databases use. MongoDB uses dynamic schemas, is cross-platform and has a document-oriented database. Hence, MongoDB can be used as a blackboard between platforms.

OpenCV Open Computer Vision can easily be integrated with ROS, it is released under the BSD license and can be used on multiple platforms. It has a focus on real-time applications and has been proven in many projects. Hence, it is logical to choose for the OpenCV library to integrated OpenCV in REXOS. The computer vision is used to identify and localise parts within the working space of the equiplet and is used for other logistic processes necessary for configuration and calibration of the systems, e.g., identification of a new gripper.

E. REXOS

REXOS has been developed using JADE and ROS 1. The main reason for using ROS is its proven use in many projects and the experience from other projects. The combination of

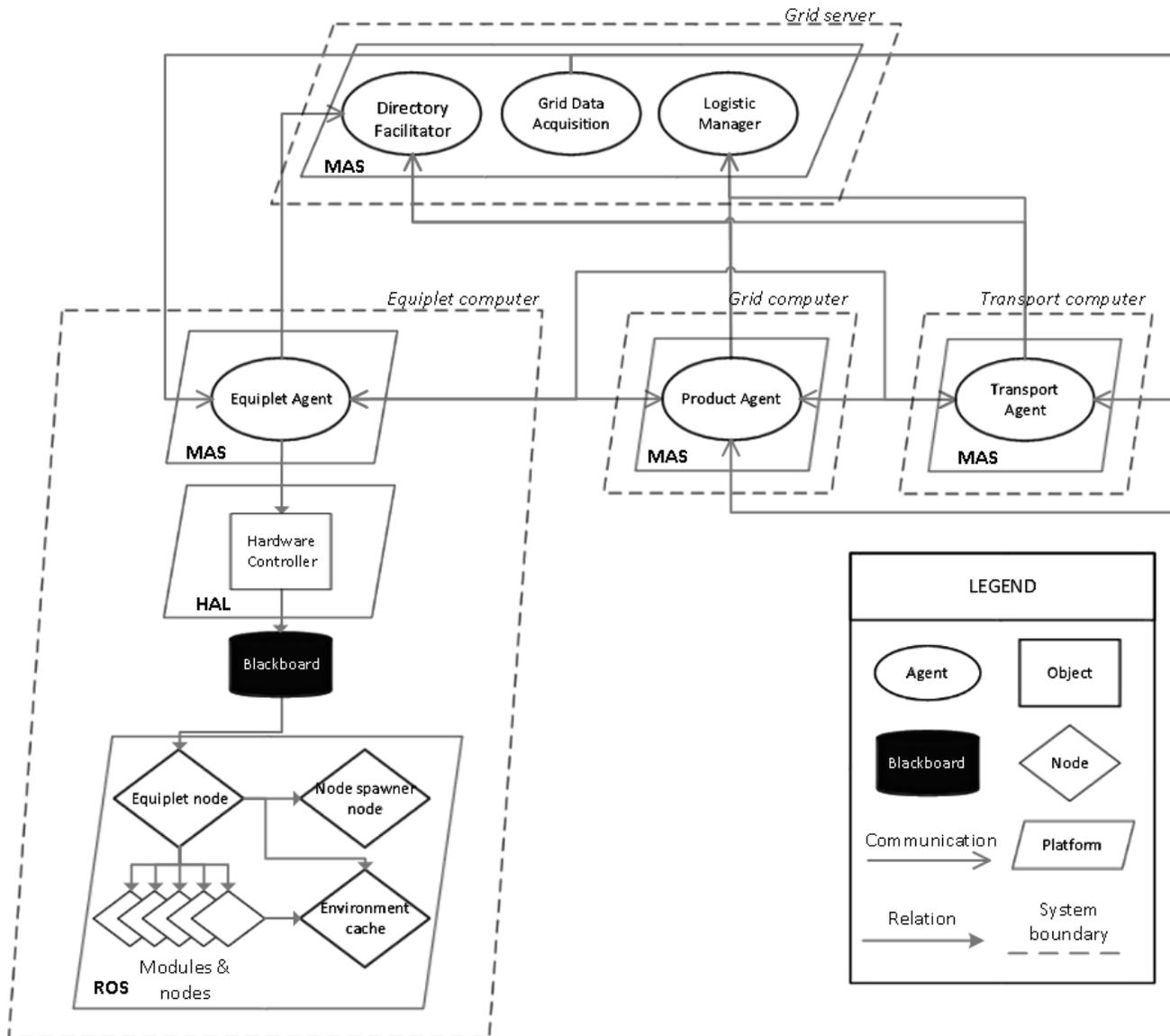


Figure 12. High-level software design of REXOS.

JADE and ROS results in the architecture as shown in Figure 12, the high-level design of REXOS.

REXOS is a distributed multi-platform system and as such will run on a number of computers. The basic logistics will run on a grid server that provides a Directory Facilitator (DF), Grid Data Acquisition System and Logistic manager. The DF can be seen as a yellow page service that knows which equiplot agents are active and what services they provide for the products. The Data acquisition will be used for statistical and Enterprise Resource Planning (ERP). The Logistic manager is mainly meant for transportation within the grid.

Products will be created dynamically, and when they are created they will usually be created by an application and then be moved to the grid server where it will be produced. If the product has an embedded computer the product agent will be moved to the product after it has been completed. However,

it might also exist in the cloud. This way the product agent can be of value throughout the entire life-cycle of the product. This way it can provide a number of services for the owner and others who use it, e.g., manuals, repair or recycle information [35].

Transport agents are responsible for the transportation of the device. Depending on the implementation this could be done in a number of ways, including Autonomous Ground Vehicles (AGV) or with (multi-directional) conveyor belts.

The equiplot will be the main system in a grid and will have a number of main platforms that each consist of one or more entities:

- 1) The equiplot agent
- 2) The Hardware Abstraction Layer (HAL)
- 3) The ROS layer

All these platforms will commonly reside on one computer

and are embedded within the equiplot. The equiplot agent will represent the equiplot hardware and interact with the grid and the products. It will also deal with scheduling and determine its capabilities based on its configuration. When a product arrives on schedule to be manufactured it will send its product steps [29] to the equiplot agent that will forward it to the Hardware Abstraction Layer (HAL). The HAL is capable to interpret the steps and translate them to specific instruction known as 'hardware steps' that will be send to the ROS layer to be executed.

The ROS layer consists of an equiplot node and at least one node per module that represents the hardware module. It also consists of a spawner node that is able to start new nodes when modules are reconfigured. The equiplot node will receive instructions from the HAL. The ROS layer uses an environment cache that represents the physical dynamic environment. Information that the environment cache holds is, for example, the position of products that are perceived by a computer vision or external system.

The interface between the different platforms is essential for a successful hybrid architecture. While Figure 12 shows a blackboard, other implementations have been developed and will be discussed in the next section.

IX. IMPLEMENTATION

For this paper, the basic architecture of REXOS is being researched. Therefore, the main focus will be on the infrastructure and platforms that are required for grid manufacturing. As mentioned before in the introduction, the hardware is an important aspect to prove the feasibility of the concept. As such, this section will show the implementation of the software platforms, the interfaces, but also give an overview of the hardware that is used.

A. middleware

Figure 13 shows the implementation of an example of the JADE platform for grid manufacturing, which consists of two equiplots and a grid server. JADE uses a main container that can be connected to remote containers (which are in the other equiplots). The main container holds a container table (CT) and two special agents, named the Agent Management Service (AMS) and the Directory Facilitator (DF). JADE has the ability to replicate or restore the main container to remain fully operational in case of a failure.

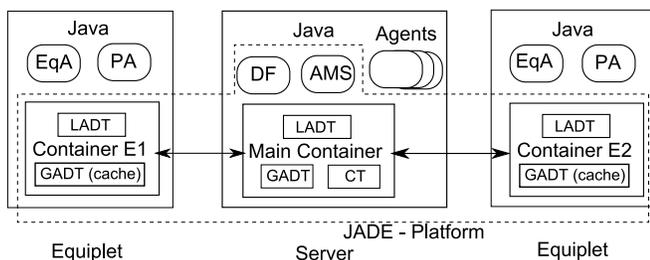


Figure 13. The Java Agent Development Platform.

In every container there is a Global agent descriptor table (GADT) that registers all the agents in the platform, including their status and location, and a local descriptor table (LADT). The GADT in the remote containers will be used for caching.

EqA and PA are the Equiplot and Product agent who will represent a specific product or equiplot.

B. Grid

The grid provides logistic functionality, which the autonomous equiplots can use. Based on the architecture shown in Figure 12 it is standard that the GRID services run on a separate server. However, since the software runs on a standard linux system and the JADE environment can be moved or distributed in any way, the GRID functionality can be run on any computer within the network. As such it is possible to start it on a computer within an equiplot. This creates the ability to quickly setup the functionality of an equiplot without requiring a complete infrastructure.

Transport [36], [37], [38] and other logistic systems like scheduling [39] are discussed in separate research and are considered out of scope for this paper.

C. Equiplots

The equiplot and its modules are specifically designed with grid manufacturing in mind. An equiplot consists of a rigid base with standard mounting points to attach modules. A standard equiplot is commonly used for assembly actions and as such typically uses 4 modules to be attached, a manipulator, gripper, vision system and a working plane. A standard equiplot stands on a rails to be easily moved and holds a standard on-board PC. Equiplots and a number of modules has been developed and tested, Figure 14 shows a demo setup of 2 equiplots configured with a pick and place setup.



Figure 14. Equiplot demo setup.

The REXOS architecture is based upon different technologies, the C++ based ROS and the JAVA based JADE platform. Therefore, the interface between these two is an important aspect for stability, performance and, therefore, scalability

issues. Due to the importance three different implementations have been developed:

- 1) Blackboard
- 2) ROS bridge
- 3) ROS Java

The *blackboard* implementation (see Figure 15) uses a MongoDB database server and multiple MongoDB database clients. The HAL and ROS components each have a client, connected to the server. By enabling the replication feature of MongoDB (usually used to keep the databases of multiple servers synchronised), the server generates an operation log, which logs all databases and collections on the server. The clients listen to this operation log using a tailable cursor. This enables the clients to communicate with each other via the server without having to periodically query the server.

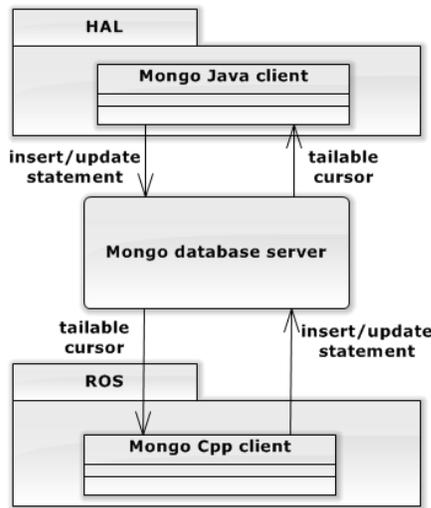


Figure 15. ROS HAL interface using a blackboard.

The *ROS bridge* implementation (see Figure 16) uses a ROS node, which acts as proxy between the HAL and ROS components. The bridge is written in Python and is designed for flexible integration of ROS in other non-C++ systems. The bridge acts as a websocket server to the outside (for REXOS this is the HAL component) and acts as a standard ROS node to the inside (for REXOS this is the ROS component). Because the bridge acts as a standard ROS node, the ROS component of REXOS can use standard ROS communication methods and messages, reducing the complexity of the interface.

The *ROS Java* implementation (see Figure 17) uses the *rosjava_core* library to communicate between the HAL and ROS components. ROS is currently available for C++ and Python. *Rosjava_core* is an attempt to make ROS available for Java. It implements the internal ROS infrastructure including time synchronisation, namespace resolving, topic and service advertising, and communication methods.

D. Modules

Modules are usually not designed specifically for a product, this enables the equilets to offer generic services to a variety

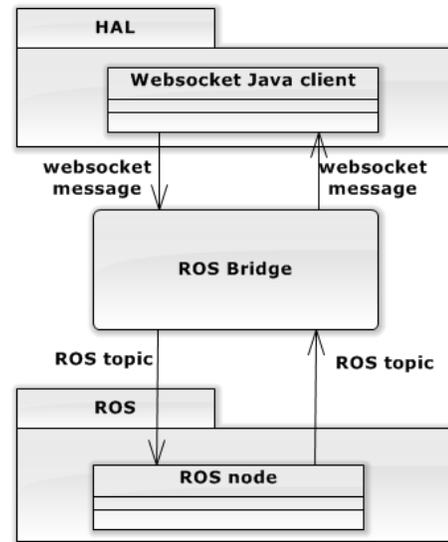


Figure 16. ROS-HAL interface using a ROS bridge.

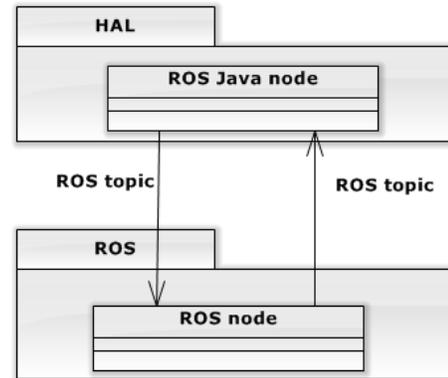


Figure 17. ROS HAL interface using a ROS JAVA node.

of products. For this purpose, a components off the shelf (COTS) strategy is adopted together with modules that are specifically designed for grid manufacturing using equilets, and which are developed using product family engineering. As shown in Figure 18, Product Family Engineering uses common parts for as many different modules as possible.

Currently, a number of modules have been developed specifically for grid manufacturing:

Delta Robot The deltarobot is a parallel manipulator where three actuators are located on the base, and where arms made of light composite material are used to move parts. All moving parts have a small inertia, which allows for very high speed and accelerations.

Figure 19 shows the schematics of the deltarobot that is specifically designed to be used for equilets. The end effector is designed in such a way that grippers can easily be changed using a precise clicking system with magnets. Many components of the delta robot are also manufactured using additive manufacturing, making it easy to customise or

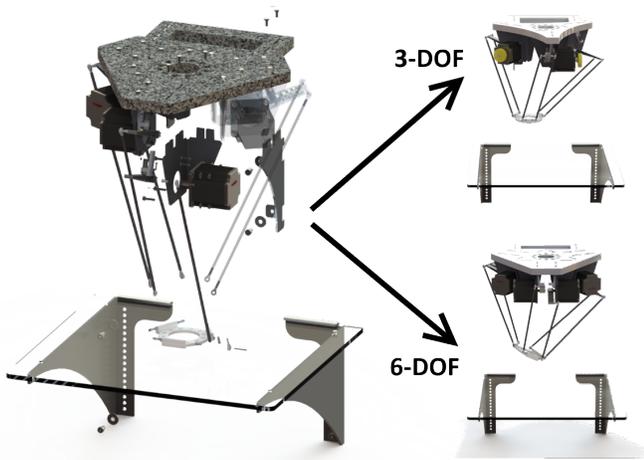


Figure 18. Two different parallel manipulators with 3 and 6 Degrees of Freedom using as many identical components as possible.

produce parts for the modules on demand.

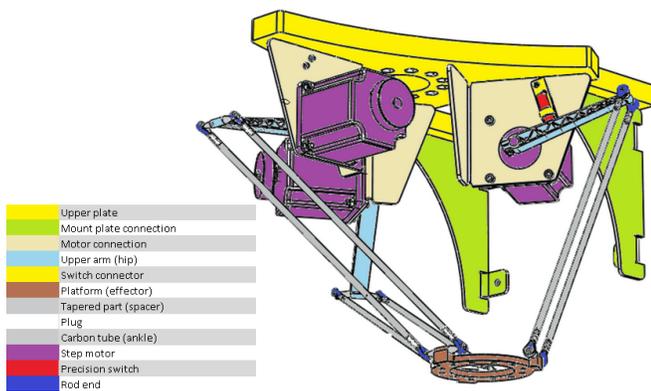


Figure 19. Delta robot Hardware Component Schematics.

The Delta Robot uses three actuators of the type Oriental Motors PK566PMB) that are controlled using motor controllers (Oriental Motors CRD514-KD). The controllers are directly accessed from the respective ROS module node. Figure 20 shows the steppermotor class, which uses a modbus interface. The modbus interface is offered by a generic InputOutput class that is implemented by the InputOutputModBusRtuController class. This class has been created for easy reuse throughout the system and is implemented by all RTU modbus implementations.

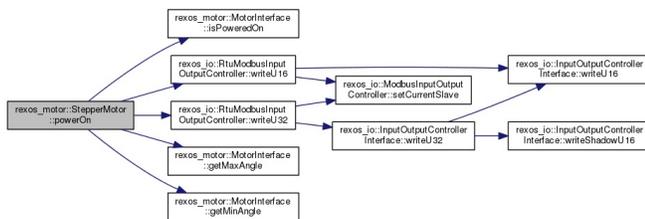


Figure 20. The call/inheritance graph for the motor controller.

Since equilets are not specifically designed for a product it is important to have as much flexibility as possible to service

a larger variety of products types. For this purpose, the Delta Robot design with 3 degrees of freedom was adapted to an inverted **Stewart Gough** platform, which uses 6 motors to be able to have a limited 6 degrees of freedom. Figure 21 shows this 6 DOF parallel manipulator module. This module is very useful since products can arrive at the equilets in any orientation.

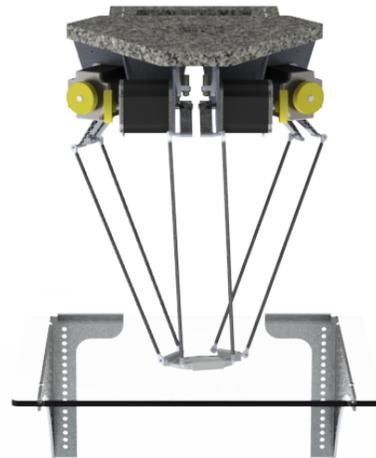


Figure 21. An adaptation to the deltarobot, which uses 6 motors as an inverted Stewart Gough Platform to create more flexibility.

This specific *Gripper* module is controlled using modbus over TCP. This is performed by an inline bus coupler (Phoenix contact IL ETH BK D18 DO4) that is accessed from the respective gripper node. Most types of grippers that are currently used work with a pneumatic system to move an effector or create a vacuum to pick up small parts.

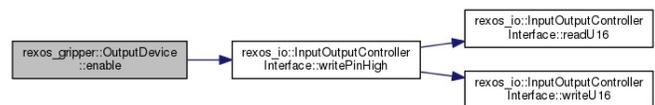


Figure 22. Gripper call / inheritance graph.

The *Vision Module* is of high importance within grid manufacturing. The product location will usually not be preprogrammed and as such must be detected dynamically in real-time. This is done by the Vision module that uses an OpenCV-based detection system to determine either the location of a product, or the location of a tray or other transport device that has the knowledge of the relative location of the product towards itself. The location data that will be found by the Vision Module will be delivered to the environment cache so that other modules can easily access it. The standard vision module holds a number of algorithms to deal with a variety of situation. It can automatically calibrate its lenses and has a built-in correction and balance system to deal with differences in lighting and distortions.

The *Work plane* is the area where a product, crate or vehicle that carries a product is located. Many equilets use transparent working planes such that computer vision systems

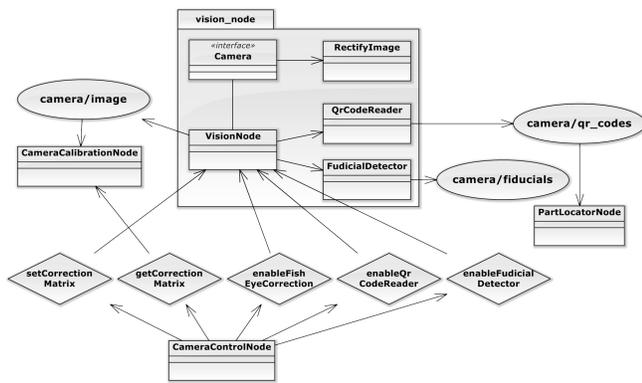


Figure 23. Vision system.

can be used to localise the specific parts. If a product is placed on a crate or cart the product agent can usually infer the location based on its own position as seen by the vision system. While the working plane has no actuators or sensors itself it is still seen as a module and has a ROS node that represents it. The ROS node is used to calculate its own position based on where it has been attached, its own known specifications and calibrations using vision and markers that can be placed on the working plane.

The *Additive Manufacturing module* can be used for a wide range of tasks. Figure 24 shows the 3D printer module that can print any object, i.e., casings or buttons for a unique customized internet radio. This module is an important asset for grid manufacturing since it makes it possible to create a variety of items not only for custom products, but also for the modules itself.

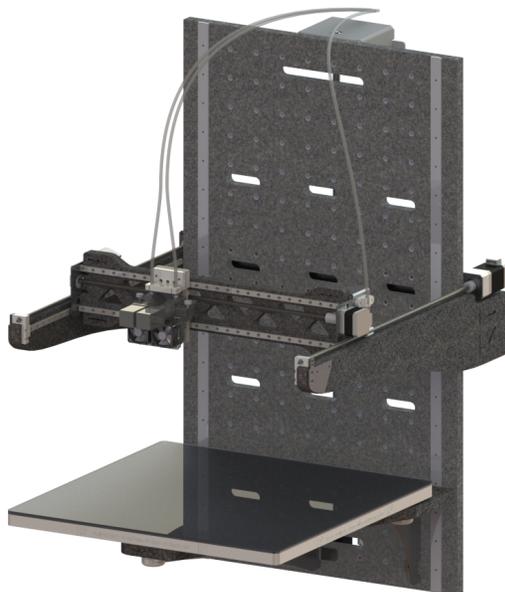


Figure 24. 3D printer module.

E. Basic operation

While not all services of a grid are relevant for this paper it is important to show the basic operation of a grid. More specific actions like reconfiguration and the implementation of the HAL will be discussed in other work. Figure 25 shows the normal operation when manufacturing. The sequence is implemented in the following way:

- 1) An equiplot agent is aware of the capabilities based on the modules it has configured.
- 2) The equiplot agent registers its service at the Directory Facilitator (DF) that acts as a 'Yellow Page' service for the Product agents.
- 3) When a product agent is initialised it has a number of product steps that describe how it needs to be manufactured, the product agent queries the DF to find the services that could potentially perform the steps.
- 4) The product agent uses the list of equiplot agents it has received from the DF to inquire the equiplot agents if the equiplot agent can match the specific schedule.
- 5) If the schedule can be met the product agent also inquires if it can meet its specific detailed criteria that the product may require for the product step to be performed adequately.
- 6) When all criteria are met and the product arrives on schedule at the equiplot it will send its instructions on how to perform the steps to the equiplot.
- 7) The equiplot will translate the steps to its specific hardware and send it to the equiplot node in ROS to control the hardware and perform the specific step.
- 8) When done the equiplot agent will inform the status to the product agent.

X. EVALUATION AND PERFORMANCE

The next step will be to evaluate the performance and scalability by performing a number of benchmarks. These have been split in multiple types:

- 1) Synthetic benchmark - to test the individual systems and latencies during load [40].
- 2) Full testing in simulation mode - to test realistic cases using the entire architecture.

A. Synthetic benchmarking

First, a standard equiplot setup has been created that uses a ROS/JADE infrastructure connected by a MongoDB blackboard, see Figure 26.

Three tests were performed:

- 1) Node to Node communication over ROS.
- 2) Agent to Agent communication using JADE.
- 3) A pick and place case utilizing all layers.

For all three benchmarks 10,000 messages will be sent, where full round time including a response message is measured. Results are shown in Figure 27, which uses a trend line over 5 periods. The message is an instruction that contains a JavaScript Object Notation (JSON) object that holds a target, ID, instruction data and parameters.

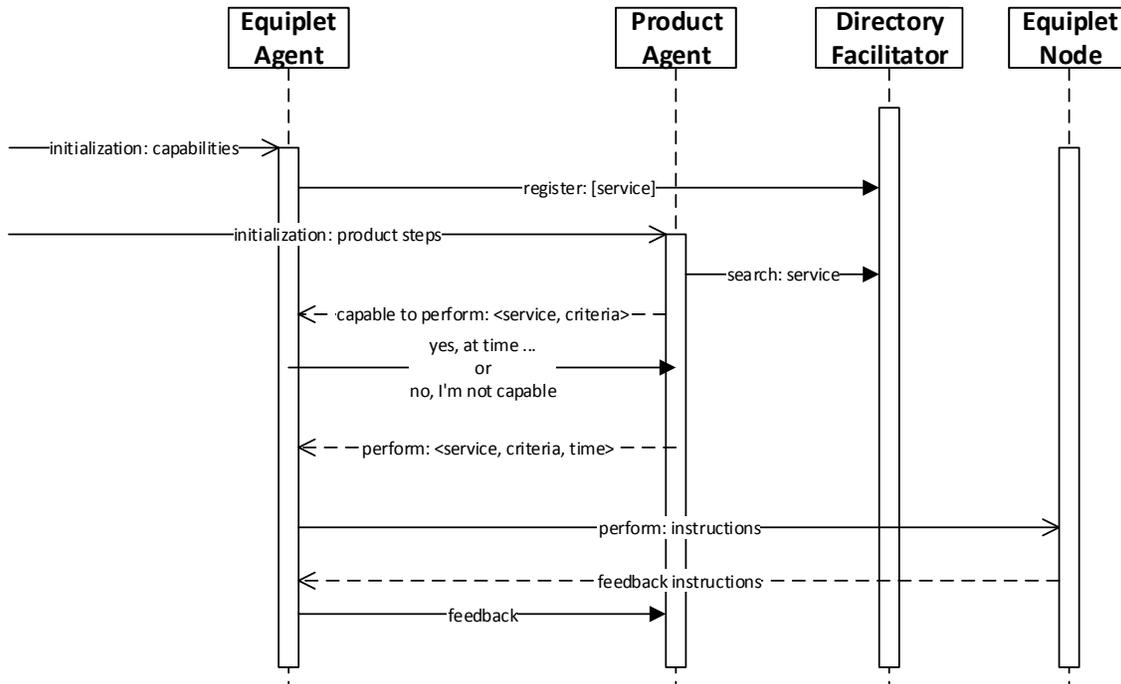


Figure 25. REXOS service.

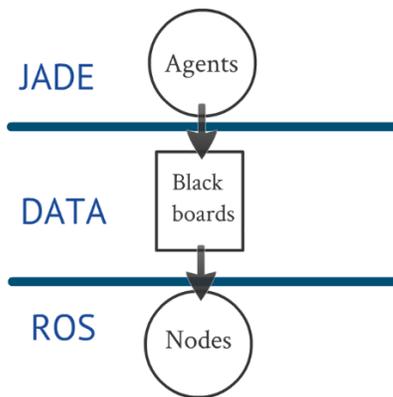


Figure 26. Synthetic test setup for benchmarking.

The results show that ROS to ROS and JADE to JADE performance is much better than when both are combined using a blackboard. Hence, different interfaces were required to be investigated to handle the communication between the ROS and agent layer. This was performed using the simulated benchmarking system.

B. Simulated benchmarking

This section evaluates the performance of the entire architecture using a full simulation of the system. The sources are identical to a real runtime situation, only the hardware responses are being simulated. The most important aspect of this test is the different interface implementations that connect the (C++ based) ROS and (JAVA based) JADE platform.

To determine the best implementation for the interface

between HAL and ROS, every implementation was benchmarked. The benchmark has been performed using custom written software and measures the time required to communicate from node A to node B and back to A. Node B will respond immediately. Node A is always a ROS node, while node B is either a regular ROS node, a ROS Java node, or a Java ROS bridge listener. The only exception are the blackboard measurements. Because the blackboard implementation does not use the ROS infrastructure, measuring the latency using ROS is not an accurate measurement. Instead the time required to communicate from A to the MongoDB server and back to A is measured. Because this gives an unfair discrepancy in the measurement (the relevant case is to transmit a message from A to B and receive and response from B), the blackboard latencies have been multiplied by two to compensate that the message has to be send twice (first from A to MongoDB and then from A to B).

The idle equiplets are equiplets that have been started but are not performing any tasks. The busy EQs are equiplets executing a hardware step every 1 second and measuring data every 10ms. The measurements have been determined using 100,000 samples.

The average latency as seen in Figure 29 has been measured with 10 and 50 active equiplets. Other scenarios have also been measured, but produced less relevant data. The ROS C++ topic, service, and action servers used are native ROS communication methods and act as a reference. They are not actual implementations of the interface. The data clearly shows that the average latency for the ROS C++ topics is the lowest. The ROS C++ service and ROS C++ actionServer also have

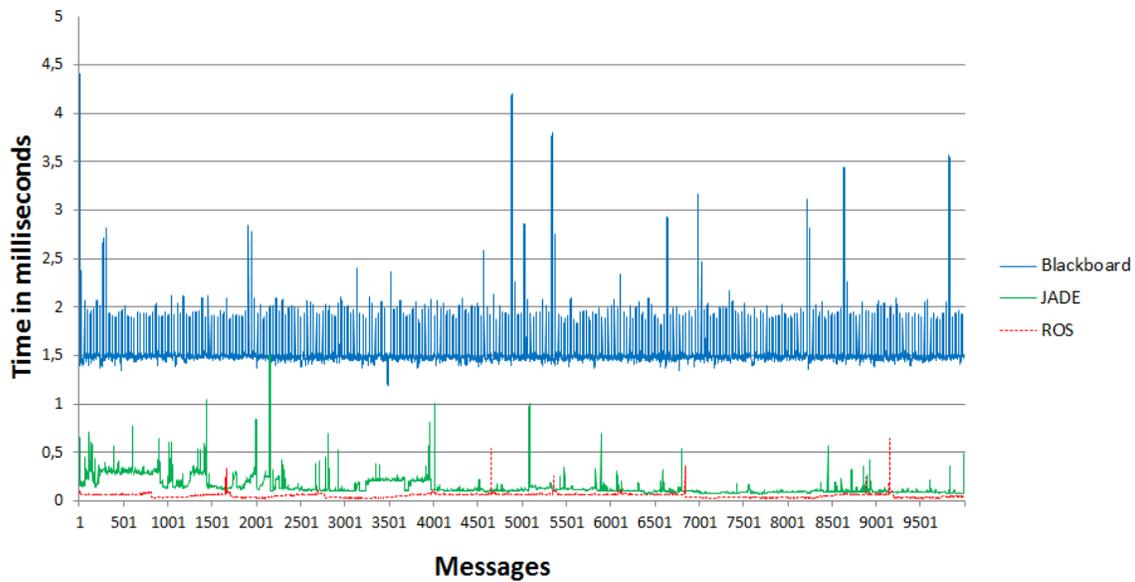


Figure 27. Synthetic benchmark of ROS to ROS communication (bottom), Agent to Agent communication (middle), and the pick and place benchmark (top).

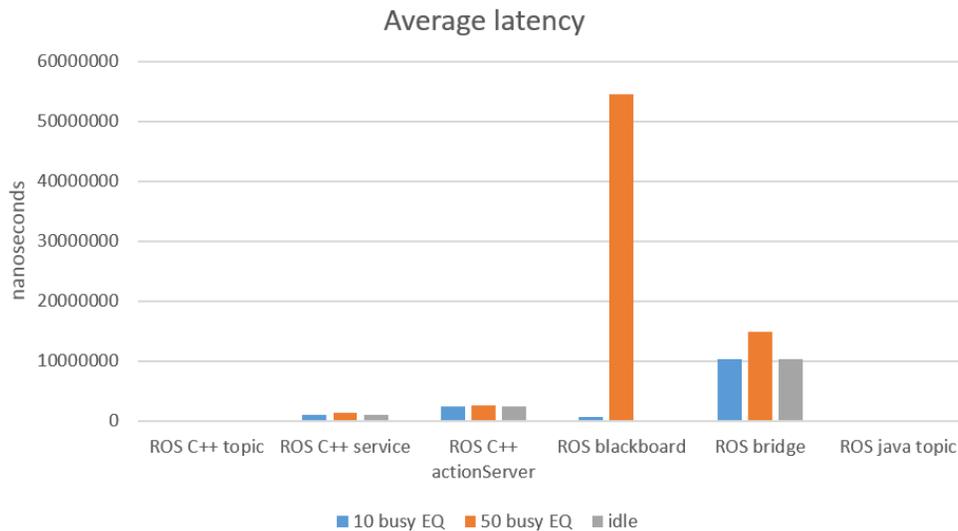


Figure 28. Average latency of the different interfaces that have been developed.

a low average latency. The blackboard implementation has a low base latency, but scales very poorly.

During the benchmarks it became clear that the blackboard implementation has a specific point after which the latency increases spectacularly. This might be caused by a connection pool in the MongoDB server running out, resulting in other connections having to wait. The ROS bridge has a very high base latency but scales much better. In all the other scenarios the base latency is also approximately 10,000,000 nanoseconds (equals 10 milliseconds). This suggests that the ROS bridge uses a periodical poll mechanism. The ROS Java topic has very low base latency and seems to scale excellently.

Figure 29 shows the consistency of the latency of an implementation of the interface. This shows how reliable the

interface is when it comes to consistent behaviour. The average deviation matches the average latency in that once again the ROS C++ topic, ROS C++ service and ROS C++ action server perform very well, while the ROS blackboard scales poorly. The ROS bridge has a quite high, but steady deviation.

XI. CONCLUSION

The paper takes an all-embracing approach to self-organising, reconfigurable autonomous manufacturing systems. The goal behind this is to provide a basis for a practical implementation by combining new technologies as a staging ground for new manufacturing methodologies based on the industry 4.0 principles that will boost the adoption by industry. Hence, this includes the development of hardware, the use of system and software engineering principles and integration of

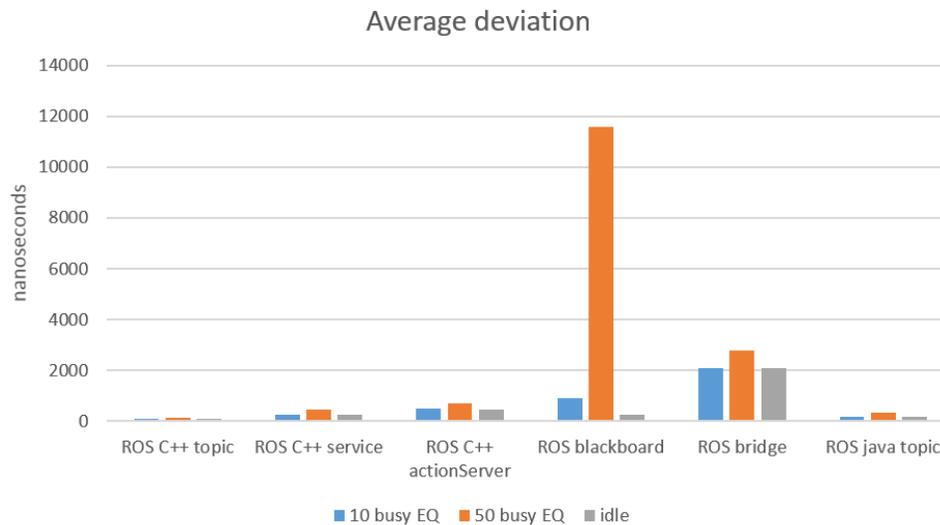


Figure 29. Average deviation of the latencies.

the newest hardware designing techniques. It also builds on current advances in software by using distributed systems and combining them in a hybrid architecture. The hope is that this leads to solutions that prove to industry that the newest technology is becoming more and more suitable for true mass adoption. This is done by investigating the current state of technology, analysing the requirements and new developments in smart industry and trying to encompass this in the concept of 'grid manufacturing' that consists of both a hardware platform, i.e., the equiplets and a software platform, i.e., REXOS.

The main research question of the paper was intended to show that grid manufacturing, based on the REXOS platform can combine low-level performance and flexibility using intelligent behaviour. The choice to combine two platforms are supported by the axiomatic design methodology, which strongly asks to decouple the requirements from the design parameters. The results give insights in how both JADE and ROS can best be interfaced using the JAVA ROS node that act as a wrapper for the messages from JADE towards the ROS platform. It also shows that the interface between these platforms are crucial to get a scalable platform, by demonstrating that the (originally developed) Blackboard interface was severely lowering the performance when 50 or more equiplets were used. However, the paper also introduces the entire concept, by introducing the GEM architecture and giving an introduction to the functionality that REXOS and the equiplets can provide.

The paper also evaluates the concept of grid manufacturing in general, taking design techniques and hardware into account. The equiplet platform in general and the modules specifically were designed using a low-cost strategy where equiplets can easily be reconfigured, providing a high utilization to a minimum cost. This was done by using combining product family engineering with the use of many standard components. When specific components have to be made they

are commonly designed in such a way that equiplets can produce them themselves, for example by using 3D printed parts.

More generally, the paper makes it clear that it is essential to take an applied approach to solve these problems. The industry will require working proof of concepts that not only tackle the theory but also the practical problems that occur when working with complex systems such as the ones demonstrated in this paper. The development and testing of all these systems have required a large amount of work but also add to the validity, and therefore, usefulness for industry.

This research provides a number of insights:

- 1) ROS and MAS can be effectively combined - which decouples the performance and intelligence gap.
 - a) The MAS provides the abstractness to deal with the dynamics that are required for self-organising systems.
 - b) ROS gives the performance and tools to effectively develop a large range of control systems that can be reconfigured.
 - c) The choice to specifically combine JADE and ROS seems to be effective.
- 2) The autonomous nature of both platforms makes it possible to adapt part of the systems during runtime, which is an important aspect when considering reconfigurability.
- 3) The use of autonomous systems makes it easier to lower interdependence between functionalities, creating a decoupled design, which lowers overall complexity.
- 4) Combining different platforms like MAS and ROS have a high potential for industry.

The combination of the requirements and propositions gives fuel to new research in more practical problems that are fundamental for smart industry; in future work both the aspects of (automatic) reconfiguration and dynamic (safety) system behaviour will also be discussed in more detail. However, the

proposition as mentioned in Section II-D seems feasible.

ACKNOWLEDGMENT

This work has been supported by the ARTEMIS R5-COP project and the HU University of Applied Sciences Utrecht, the Netherlands. Thanks go to all the computer engineering students that have helped with the implementations of REXOS, Maarten Dinkla for performing a language review, and Joost van Duijn for providing information about the hardware design process.

REFERENCES

- [1] D. Telgen, L. van Moergestel, E. Puik, P. Muller, and J.-J. Meyer, "Requirements and matching software technologies for sustainable and agile manufacturing systems," in *INTELLI 2013, The Second International Conference on Intelligent Systems and Applications*, 2013, pp. 30–35.
- [2] D. Telgen, L. van Moergestel, E. Puik, and J.-J. Meyer, "Agile manufacturing possibilities with agent technology," in *Proceedings of 22nd International Conference on Flexible Automation and Intelligent Manufacturing*. Tampere University of Technology, 2012, pp. 341–346.
- [3] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, Oct. 2009.
- [4] K. Schild and S. Bussmann, "Self-organization in manufacturing operations," *Commun. ACM*, vol. 50, no. 12, pp. 74–79, Dec. 2007.
- [5] J. Wind and A. Rangaswamy, "Customerization: The next revolution in mass customization," *Journal of Interactive Marketing*, vol. 15, no. 1, pp. 13 – 32, 2001.
- [6] K. H. Hall, R. J. Staron, and P. Vrba, "Experience with holonic and agent-based control systems and their adoption by industry," in *Holonic and Multi-Agent Systems for Manufacturing*, ser. Lecture Notes in Computer Science, V. Mak, R. William Brennan, and M. Pchouek, Eds. Springer Berlin Heidelberg, 2005, vol. 3593, pp. 1–10.
- [7] V. Marik and D. McFarlane, "Industrial adoption of agent-based technologies," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 27–35, 2005.
- [8] S. HU, "Paradigms of manufacturing - a panel discussion," *3rd Conference on Reconfigurable Manufacturing*, 2005.
- [9] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of Manufacturing Systems*, vol. 29, no. 4, pp. 130 – 141, 2010.
- [10] M. Mehrabi, A. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems: Key to future manufacturing," *Journal of Intelligent Manufacturing*, vol. 11, no. 4, pp. 403–419, 2000.
- [11] K. Stecke, "Flexibility is the future of reconfigurability. paradigms of manufacturinga panel discussion," in *3rd Conference on Reconfigurable Manufacturing*, 2005.
- [12] H. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *International Journal of Flexible Manufacturing Systems*, vol. 17, no. 4, pp. 261–276, 2005.
- [13] A. Gunasekaran, "Agile manufacturing: a framework for research and development," *International journal of production economics*, vol. 62, no. 1, pp. 87–105, 1999.
- [14] S. L. Koh and L. Wang, "Overview of enterprise networks and logistics for agile manufacturing," in *Enterprise Networks and Logistics for Agile Manufacturing*, L. Wang and S. L. Koh, Eds. Springer London, 2010, pp. 1–10.
- [15] U. Rauschecker, D. Stock, M. Stöhr, and A. Verl, "Connecting factories and related it environments to manufacturing clouds," *International Journal of Manufacturing Research*, vol. 9, no. 4, pp. 389–407, 2014.
- [16] D. C. Mcfarlane and S. Bussmann, "Developments in holonic production planning and control," *Production Planning & Control*, vol. 11, no. 6, pp. 522–536, 2000.
- [17] A. Giret and V. Botti, "Engineering holonic manufacturing systems," *Computers in Industry*, vol. 60, no. 6, pp. 428 – 440, 2009, collaborative Engineering: from Concurrent Engineering to Enterprise Collaboration.
- [18] N. P. Suh, *Axiomatic Design: Advances and Applications (The Oxford Series on Advanced Manufacturing)*. Oxford University Press, USA, 2001.
- [19] M. Wooldridge and N. Jennings, "Intelligent agents: Theory and practice," *The Knowledge Engineering Review*, vol. 10, pp. 115–152, 1995.
- [20] M. Paolucci and R. Sacile, *Agent-based manufacturing and control systems : new agile manufacturing solutions for achieving peak performance*. Boca Raton, Fla.: CRC Press, 2005.
- [21] D. Dennett, *The Intentional Stance*. Cambridge, Mass: MIT Press, 1987.
- [22] M. Bratman, *Intention, Plans, and Practical Reason*. Cambridge, Mass: Harvard University Press, 1987.
- [23] A. S. Rao, M. P. Georgeff *et al.*, "Bdi agents: From theory to practice." in *ICMAS*, vol. 95, 1995, pp. 312–319.
- [24] M. Wooldridge, *An Introduction to MultiAgent Systems, Second Edition*. Sussex, UK: Wiley, 2009.
- [25] A. Ricci, M. Viroli, and M. Piuanti, "Formalising the environment in mas programming: A formal model for artifact-based environments," in *Programming Multi-Agent Systems*, ser. Lecture Notes in Computer Science, L. Braubach, J.-P. Briot, and J. Thangarajah, Eds. Springer Berlin Heidelberg, 2010, vol. 5919, pp. 133–150.
- [26] F. Heintz, P. Rudol, and P. Doherty, "Bridging the sense-reasoning gap using dyknow: A knowledge processing middleware framework," in *KI 2007: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, J. Hertzberg, M. Beetz, and R. Englert, Eds. Springer Berlin Heidelberg, 2007, vol. 4667, pp. 460–463.
- [27] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.
- [28] N. R. Jennings, "An agent-based approach for building complex software systems," *Commun. ACM*, vol. 44, no. 4, pp. 35–41, apr 2001.
- [29] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "Decentralized autonomous-agent-based infrastructure for agile multiparallel manufacturing," *Proceedings of the International Symposium on Autonomous Distributed Systems (ISADS 2011) Kobe, Japan*, pp. 281–288, 2011.
- [30] L. van Moergestel, E. Puik, D. Telgen, and J.-J. Meyer, "Embedded autonomous agents in products supporting repair and recycling," *Proceedings of the International Symposium on Autonomous Distributed Systems (ISADS 2013) Mexico City*, pp. 67–74, 2013.
- [31] M. Dastani, "2apl: a practical agent programming language," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 3, pp. 214–248, 2008.
- [32] F. Bellifemine, C. G., and D. Greenwood, *Developing multi-agent systems with Jade*. John Wiley & Sons Ltd., 2007.
- [33] L. Braubach, A. Pohkahr, and W. Lamersdorf, "Jadex: A short overview," *Proceedings of the Main Conference Net.ObjectDays*, 2004.
- [34] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
- [35] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "The role of agents in the lifecycle of a product," *CMD 2010 proceedings*, pp. 28–32, 2010.
- [36] L. van Moergestel, E. Puik, D. Telgen, M. Kuijl, B. Alblas, J. Koelewijn, J.-J. Meyer *et al.*, "A simulation model for transport in a grid-based manufacturing system," in *Proc. of the Third International Conference on Intelligent Systems and Applications (INTELLI 2014)*. IARIA, 2014, pp. 1–7.
- [37] L. v. Moergestel, E. Puik, D. Telgen, J.-J. Meyer *et al.*, "A study on transport and load in a grid-based manufacturing system," *International Journal on Advances in Software*, vol. 8, no. 1 & 2, pp. 27–37, 2015.
- [38] L. v. Moergestel, J.-J. Meyer, E. Puik, and D. Telgen, "Agent-based manufacturing in a production grid: Adapting a production grid to the production paths," *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART 2014)*, vol. 1, pp. 342–349, 2014.
- [39] L. van Moergestel, E. Puik, D. Telgen, and J.-J. Meyer, "Production scheduling in an agile agent-based production grid," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, vol. 2, 2012, pp. 293–298.
- [40] D. Telgen, L. van Moergestel, E. Puik, P. Muller, A. Groenewegen, D. van der Steen, D. Koole, P. de Wit, A. van Zanten, and J.-J. Meyer, "Combining performance and flexibility for rms with a hybrid architecture," in *Advances in Artificial Intelligence, 16th Portuguese Conference on Artificial Intelligence, local proceedings*. IEEE, 2013, pp. 388–399.

On-Screen Point-of-Regard Estimation Under Natural Head Movement for a Computer with Integrated Webcam

Stefania Cristina, Kenneth P. Camilleri
Department of Systems and Control Engineering
University of Malta, Malta
Email: stefania.cristina@um.edu.mt
kenneth.camilleri@um.edu.mt

Abstract—Recent developments in the field of eye-gaze tracking by videoculography indicate a growing interest towards unobtrusive tracking in real-life scenarios, a new paradigm referred to as pervasive eye-gaze tracking. Among the challenges associated with this paradigm, the capability of a tracking platform to integrate well into devices with in-built imaging hardware and to permit natural head movement during tracking is of importance in less constrained scenarios. The work presented in this paper builds on our earlier work, which addressed the problem of estimating on-screen point-of-regard from iris center movements captured by an integrated camera inside a notebook computer, by proposing a method to approximate the head movements in conjunction with the iris movements in order to alleviate the requirement for a stationary head pose. Following iris localization by an appearance-based method, linear mapping functions for the iris and head movement are computed during a brief calibration procedure permitting the image information to be mapped to a point-of-regard on the monitor screen. Following the calculation of the point-of-regard as a function of the iris and head movement, separate Kalman filters improve upon the noisy point-of-regard estimates to smoothen the trajectory of the mouse cursor on the monitor screen. Quantitative and qualitative results obtained from two validation procedures reveal an improvement in the estimation accuracy under natural head movement, over our previous results achieved from earlier work.

Keywords—Point-of-regard estimation; Eye-gaze tracking; Iris center localization; Head movement tracking; Integrated webcam.

I. INTRODUCTION

The idea of estimating the human eye-gaze has been receiving increasing interest since at least the 1870s, following the realization that the eye movements hold important information that relates to visual attention. Throughout the years, efforts in improving eye-gaze tracking devices to minimize discomfort and direct contact with the user led to the conception of videoculography (VOG), whereby the eye movements are tracked remotely from a stream of images that is captured by digital cameras. Eye-gaze tracking by VOG quickly found its way into a host of applications, ranging from human-computer interaction (HCI) [1]–[3], to automotive engineering [4][5]. Indeed, with the advent of the personal computer, eye-gaze tracking technology was identified as an alternative controlling medium enabling the user to operate the mouse cursor using the eye movements alone [3].

Following the emergence and widespread use of highly mobile devices with integrated imaging hardware, there has been an increasing interest in mobile eye-gaze tracking that blends well into the daily life setting of the user [6]. This emerging interest led to the conception of a new paradigm that is referred to as *pervasive tracking*, which refers to the endeavor of tracking the eye movements continuously in different real-life scenarios [6]. This notion of pervasive eye-gaze tracking is multi-faceted, typically characterized by different aspects such as the capability of a tracking platform to permit tracking inside less constrained conditions, to track the user remotely and unobtrusively and to integrate well into devices that already comprise imaging hardware without necessitating hardware modification.

Nevertheless, this new paradigm brings challenges that go beyond the typical conditions for which classical video-based eye-gaze tracking methods have been developed. Despite considerable advances in the field of eye-gaze tracking as evidenced by an abundance of methods proposed over the years [7], video-based eye-gaze tracking has been mainly considered as a desktop technology, often requiring specific conditions to operate. Commercially available eye-gaze tracking systems, for instance, are usually equipped with high-grade cameras and actively project infra-red illumination over the face and the eyes to obtain accurate eye movement measurements. In utilizing specialized hardware to operate, active eye-gaze tracking fails to integrate well into devices that already comprise imaging hardware, while its usability is constrained to controlled environments away from interfering infra-red sources. On the other hand, passive eye-gaze tracking that operates via standard imaging hardware and exploits the appearance of the eye without relying on specialized illumination sources for localization and tracking, provides a solution that promises to integrate better into pervasive scenarios.

Nonetheless, utilizing existing passive eye-gaze tracking methods to address the challenges associated with pervasive tracking, such as the measurement of eye movement from low-quality images captured by lower-grade hardware, may not necessarily be a suitable solution. For instance, existing shape-based methods that localize the eye region inside an image frame by fitting curves to its contours, often require images of suitable quality and good contrast in which the boundaries between different components such as the eyelids, the sclera and the iris are clearly distinguishable [8]–[11]. Similarly,

feature-based methods that search for distinctive features such as the limbus boundary [12][13], necessitate these features to be clearly identifiable. Appearance-based methods relying on a trained classifier, such as a Support Vector Machine (SVM), to estimate the 2D point-of-regard directly from an eye region image without identifying its separate components, have been reported to perform relatively well on lower-quality images as long as the training data includes images of similar quality as well [14]. However, this performance usually comes at the cost of lengthy calibration sessions that serve to gather the user-dependent data that is required for training [15][16].

Moreover, recent attempts to track the eye-gaze on mobile platforms by existing eye-gaze tracking methods [17][18] have reported undesirable constraints such as a requirement for close-up eye region images [19], lengthy calibration sessions [17] and stationary head poses [15][20][21] often with the aid of a chin-rest [15][20]. The capability of a tracking platform to allow head movement during gaze estimation, in particular, is an important aspect in the context of pervasive eye-gaze tracking, permitting the user to move naturally without constraining the tracking conditions. Methods that do not cater for head movement during tracking and calibration often estimate an image-to-screen mapping function that is valid for a single head pose alone, hence requiring a stationary head pose during and following calibration [15][20][21]. While this may be considered as a workable solution for short-term use, it does not provide for a comfortable setup over longer periods of time. Head movement compensation is, therefore, required in order to lift the constraint of maintaining a stationary head pose, permitting small head displacement if the movement is measured by a simple head marker [22][23] or higher degrees-of-freedom if the head pose is calculated in 3-dimensional space [9][24][25]. In this regard, several eye-gaze tracking methods that cater for head movement via an appearance-based approach generally populate a training dataset with images captured under different head orientations [26][27], often at the expense of presenting the user with a larger set of calibration targets resulting in a data collection session that is considerably prolonged [26]. Feature-based approaches for head pose estimation, on the other hand, generally follow a model-fitting approach whereby a face model is fit to specific face feature landmarks allowing the estimation of the head rotation angles [19][28][29]. The accuracy of these approaches is often contingent upon accurate tracking of several facial features, which is in turn susceptible to feature distortion and self-occlusion during head rotations [19].

In light of the challenges associated with pervasive tracking, we propose a passive eye-gaze tracking method to estimate the point-of-regard (POR) on a monitor screen from lower-quality images acquired by an integrated camera inside a notebook computer, while approximating any natural, typically small, head rotations performed by the user during tracking. To localize the iris center coordinates from low-resolution eye region images while the user sits at a distance from the monitor screen, we propose an appearance-based method that localizes the iris region by its intensity values. In addition, our method ensures that the iris region can be located at different angles of eyeball and head rotation and under partial occlusion by the

eyelids, and can be automatically relocated after this has been entirely occluded during blinking. Following iris localization, the iris center coordinates extracted earlier are mapped to a POR on the monitor screen via linear mapping functions that are estimated through a brief calibration procedure. Linear mapping functions for the head movement are also estimated during calibration, ensuring that any natural head rotations performed by the user during tracking are handled according to the resulting face region displacement inside the image space. Kalman filters handling the iris and head movement separately are assigned to each of four screen quadrants, characterised by different mapping functions, in order to improve upon the noisy POR estimates computed as a function of the iris and head movement. This serves to smoothen the trajectory of the mouse cursor on the monitor screen.

The details of the proposed passive eye-gaze tracking method are described in Section II. Section III presents and discusses the experimental results. A comparison between the results achieved by our method and those reported by relevant state-of-the-art methods is provided in Section IV, while Section V draws the final remarks that conclude the paper.

II. METHOD

The following sections describe the stages of the proposed method, starting off with eye region detection and tracking up to the estimation of the POR onto the monitor screen.

A. Eye Region Detection

The estimation of the POR on the monitor screen requires that the eye region is initially detected inside the first few image frames. Searching for the eye region over an entire image frame can be computationally expensive for a real-time application and can lead to the occurrence of several false positive detections. Therefore, prior to detecting the eye region, the bounding box that encloses the face region is detected first such that this constrains the search range for the eye region, reducing the searching time as well as the possibility of false positives. The eye region is subsequently detected within the area delimited by the boundaries of the face region.

Given the real-time nature of our application, we chose the Viola-Jones algorithm for rapid detection of the face and eye region [30]. Within the Viola-Jones framework, features of interest are detected by sliding rectangular windows of Haar-like operators over an image frame, subtracting the underlying image pixels that fall within the shaded regions of the Haar-like operators from the image pixels that fall within the clear regions. Candidate image patches are classified between positive and negative samples by a cascade of weak classifiers arranged in order of increasing complexity. Every weak classifier is trained to search for a specific set of Haar features by a technique called boosting, such that each stage processes the samples that pass through the preceding classifier and rejects the negative samples as early into the cascade as possible to ensure computational efficiency.

The face and eye region detection stages in our work utilize freely available cascades of classifiers that come with the OpenCV library [31], which had been previously trained on a wide variety of training images such that detection generalizes well across different users. Since the training data for these classifiers was mainly composed of frontal face and eye region samples, the user is required to hold a frontal head pose for a brief period of time until the face and eye regions have been successfully detected. In case multiple candidates are detected by the face region classifier, the proposed method chooses the candidate that is closest to the monitor screen characterized by the largest bounding box, and discards the others.

B. Eye Region Tracking

To allow for small and natural head movement during tracking without requiring the uncomfortable use of a chinrest, the initial position of the eye region detected earlier needs to be updated at every image frame to account for its displacement in the x - and y -directions. While performing eye region detection on a frame-by-frame basis would be a possible solution to estimate the eye region displacement through an image sequence, such an approach would be sub-optimal in terms of computational efficiency for a real-time application. Therefore, assuming gradual and small head displacement, the eye region is tracked between successive image frames by template matching, using the last known position of the eye region inside the previous image frame to constrain the search area inside the next frame.

A template image of the eye region is captured and stored following earlier detection of this region by the Viola-Jones algorithm. The template image is then matched to the search image inside a window of fixed size, centered around the last known position of the feature of interest. Template matching utilizes the normalized sum of squared differences (NSSD) as a measure of similarity, denoted as follows,

$$NSSD(x, y) = \frac{\sum_{x', y'} [T(x', y') - I(x + x', y + y')]^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \sum_{x', y'} I(x + x', y + y')^2}} \quad (1)$$

where T denotes the template image and I denotes the search image. A NSSD value of zero represents a perfect match between the template and search image, whereas a higher value denotes increasing mismatch between the two images. This permits the identification of the new position of the feature of interest, which is specified by the location inside the search image that gives the minimum NSSD value after template matching.

C. Iris Center Localization

The movement of the eyes is commonly represented by the trajectory of the iris or pupil center in a stream of image frames [7], and hence the significance of localizing the iris or pupil center coordinates after the eye region has been detected. Given the small footprint of the eye region inside the image space,

we opt to localize the iris center coordinates rather than the pupil, since the iris occupies a larger area inside the eye region and can be detected more reliably.

While there exist different methods that permit localization of the iris region inside an image frame, not all of these methods are suitable for localizing the iris region from low-resolution images, especially if fine details such as the contours of different components of the eye [8]–[11] need to be clearly distinguishable. We propose an appearance-based method that segments the iris region via a Bayes' classifier to localize it. The Bayes' classifier is trained during an offline training stage to classify between iris and non-iris pixels based on their red channel value in the RGB color space. During tracking, intensity values of pixels residing within the eye region are classified as belonging to the iris region if their likelihood exceeds a pre-defined threshold value, θ :

$$\frac{p(x_r(i, j) | \varpi_{iris})}{p(x_r(i, j) | \varpi_{non-iris})} \geq \theta \quad (2)$$

where $p(x_r(i, j) | \varpi_{iris})$ denotes the class-conditional probability of observing a red-band measurement at pixel (i, j) knowing it belongs to the iris class, while $p(x_r(i, j) | \varpi_{non-iris})$ denotes the class-conditional probability of observing the same red-band measurement at pixel (i, j) knowing it belongs to the non-iris class. The resulting binary image contains a blob of pixels that belongs to the iris region, whose center of mass is taken to represent the iris center coordinates. In case the eyebrow is also mistakenly classified as belonging to the iris region due to the resemblance in color with dark irises, the blob of pixels that is closer to the center of the eye region is considered to represent the iris.

The Bayes' classifier had been previously used for skin region segmentation in images [32], but to our knowledge it has never been adopted to the problem of iris region localization for eye-gaze tracking until our work of [1]. Preliminary results have shown this method to be suitable in localizing the iris region from low-quality images, owing especially to the fact that the proposed localization method depends upon statistical color modeling rather than geometrical information. Another advantage that is also related to its independency from geometrical information is the ability to locate the iris region at different angles of rotation and under partial occlusion by the eyelids. The main downside of this method is its susceptibility to illumination variations, which problem is however alleviated by training the Bayes' classifier on iris and non-iris pixels acquired under different illumination conditions.

D. POR Estimation

Having determined the iris center coordinates, the final stage seeks to estimate the user's POR on the monitor screen from the iris displacement while approximating any head rotations performed by the user during tracking.

For simplicity, and since we expect small eyeball and head rotation angles within the width and height of the monitor screen at close range, the eyeball and head rotations performed

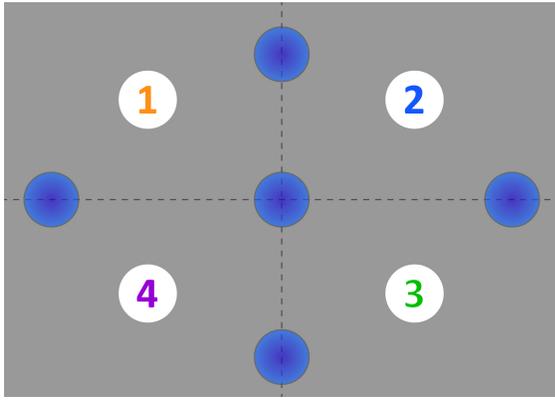


Figure 1. Strategically placed calibration points divide the screen display into four separate quadrants.

by the user during tracking are approximated by planar displacement of the respective iris center and face region inside the image space. The on-screen displacement, $\Delta \mathbf{x}_{s_g}$, of the user's POR may therefore be defined as a function of the iris displacement, $\Delta \mathbf{x}_{s_e}$, and head displacement, $\Delta \mathbf{x}_{s_h}$, on the monitor screen as follows,

$$\Delta \mathbf{x}_{s_g} = \Delta \mathbf{x}_{s_e} + \Delta \mathbf{x}_{s_h} \quad (3)$$

In order to estimate the on-screen POR displacement, we seek to determine image-to-screen mapping relationships for the iris and head components. The assumption of planar iris center displacement permits us to define a linear mapping relationship between the image and screen coordinates as follows,

$$(\mathbf{x}_{s_e}^{(3)} - \mathbf{x}_s^{(1)}) = \frac{(\mathbf{x}_s^{(2)} - \mathbf{x}_s^{(1)})}{(\mathbf{x}_{i_e}^{(2)} - \mathbf{x}_{i_e}^{(1)})} (\mathbf{x}_{i_e}^{(3)} - \mathbf{x}_{i_e}^{(1)}) \quad (4)$$

where $\mathbf{x}_s^{(1)}$ and $\mathbf{x}_s^{(2)}$ denote the screen coordinates of two calibration points respectively, whereas $\mathbf{x}_{i_e}^{(1)}$ and $\mathbf{x}_{i_e}^{(2)}$ denote the corresponding iris center coordinates inside the eye region which are estimated while the user fixates at the two calibration points maintaining the head stationary. During tracking, the mapping function in (4) computes the displacement in screen coordinates between the new POR $\mathbf{x}_{s_e}^{(3)}$ and the calibration point $\mathbf{x}_s^{(1)}$, following the estimation of the displacement in image coordinates between the new iris center location $\mathbf{x}_{i_e}^{(3)}$ and the previously estimated $\mathbf{x}_{i_e}^{(1)}$. In order to compensate for the assumption of planar iris movement, the monitor screen is divided into four separate quadrants by strategically placed calibration points as illustrated in Figure 1, such that each quadrant is assigned different parameter values that best describe the linear mapping between the image-to-screen coordinates.

Similarly, a linear image-to-screen mapping relationship for the head displacement is defined as follows,

$$(\mathbf{x}_{s_h}^{(3)} - \mathbf{x}_s^{(1)}) = \frac{(\mathbf{x}_s^{(2)} - \mathbf{x}_s^{(1)})}{(\mathbf{x}_{i_h}^{(2)} - \mathbf{x}_{i_h}^{(1)})} (\mathbf{x}_{i_h}^{(3)} - \mathbf{x}_{i_h}^{(1)}) \quad (5)$$

requiring the estimation of image coordinates, $\mathbf{x}_{i_h}^{(1)}$ and $\mathbf{x}_{i_h}^{(2)}$, while the user fixates at the corresponding on-screen calibration targets, $\mathbf{x}_s^{(1)}$ and $\mathbf{x}_s^{(2)}$, by the head pose alone maintaining the eyes stationary. In order to reduce the calibration effort and re-use the iris mapping relationship estimated earlier for the visual targets in Figure 1, the user is requested to perform head rotations in the horizontal and vertical directions while fixating at a single calibration target. Since the user's POR on the monitor screen is maintained fixed during the head movement, Equation (3) reduces to,

$$\Delta \mathbf{x}_{s_e} + \Delta \mathbf{x}_{s_h} = 0 \quad (6)$$

such that the on-screen relationship between the iris and head displacement may be defined as follows,

$$\Delta \mathbf{x}_{s_e} = -\Delta \mathbf{x}_{s_h} \quad (7)$$

Inside the image space, $\Delta \mathbf{x}_{i_e} \propto \Delta \mathbf{x}_{i_h}$ up to a scale factor S which models the relationship between the different rotation radii of the eyeball and the head around their respective axis. As the user performs arbitrary head movement during calibration, tuples of coordinates \mathbf{x}_{i_e} and \mathbf{x}_{i_h} are collected allowing the scale factors S_x and S_y , in the horizontal and vertical directions, to be estimated as follows,

$$S_x = \frac{\Delta x_{i_h}}{\Delta x_{i_e}} \quad S_y = \frac{\Delta y_{i_h}}{\Delta y_{i_e}} \quad (8)$$

where displacements, $\Delta \mathbf{x}_{i_e}$ and $\Delta \mathbf{x}_{i_h}$, are estimated with respect to initial eye and head poses. Equation (5) may hence be re-defined as,

$$(\mathbf{x}_{s_h}^{(3)} - \mathbf{x}_s^{(1)}) = \frac{(\mathbf{x}_s^{(2)} - \mathbf{x}_s^{(1)})}{\mathbf{S} \circ (\mathbf{x}_{i_e}^{(2)} - \mathbf{x}_{i_e}^{(1)})} (\mathbf{x}_{i_h}^{(3)} - \mathbf{S} \circ \mathbf{x}_{i_e}^{(1)}), \quad (9)$$

$$\mathbf{S} = \begin{pmatrix} S_x \\ S_y \end{pmatrix}$$

where $\mathbf{x}_{s_h}^{(3)}$ denotes the new POR determined by the head movement alone, corresponding to a new head region location $\mathbf{x}_{i_h}^{(3)}$ inside the image space. Having determined the iris, $\Delta \mathbf{x}_{s_e}$, and head displacement, $\Delta \mathbf{x}_{s_h}$, following Equations (4) and (9), respectively, the on-screen POR displacement may be estimated from Equation (3).

To alleviate the issue of noisy iris center and head displacement measurements from low-quality images, and hence smoothen the trajectory of the mouse cursor on the monitor screen after mapping the iris and head movement to a POR, we propose to employ Kalman filtering to improve upon these noisy measurements. Indeed, the Kalman filter is an algorithm that recursively utilizes noisy measurements observed over time to produce estimates of desired variables that tend to be more accurate than the single measurements alone [33]. We define separate Kalman filters to handle the respective iris and

head displacement for our specific application of smoothing the mouse cursor trajectory as follows:

State Vector: We define the state vector $\mathbf{x}_{k+1}^{(e)}$ as,

$$\mathbf{x}_{k+1}^{(e)} = [\Delta x_{s_e} \quad \Delta y_{s_e}]^T_{k+1} \quad (10)$$

comprising the horizontal iris centre displacement, $\Delta x_{s_e} = (x_{s_e}^{(3)} - x_s^{(1)})$, and similarly for the vertical iris centre displacement displacement, Δy_{s_e} . Similarly, for the other Kalman filter, we define the state vector $\mathbf{x}_{k+1}^{(h)}$ as,

$$\mathbf{x}_{k+1}^{(h)} = [\Delta x_{s_h} \quad \Delta y_{s_h}]^T_{k+1} \quad (11)$$

where Δx_{s_h} and Δy_{s_h} denote the horizontal and vertical on-screen head displacement.

Transition Matrix: Assuming the eye and head movement during tracking to consist of fixation periods and smooth movement between one visual stimulus and another, we represent the transition matrices $A^{(e)}$ and $A^{(h)}$ by simple linear models of the ideal mouse cursor trajectory during constant velocity movement as follows,

$$A^{(e)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (12)$$

and,

$$A^{(h)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (13)$$

Measurement Vector: In our work, the measurement vector $\mathbf{z}_{k+1}^{(e)}$, attributed to the Kalman filter that handles the iris movement, holds the estimated image displacement of the iris center coordinates as follows,

$$\mathbf{z}_{k+1}^{(e)} = [\Delta x_{i_e} \quad \Delta y_{i_e}]^T_{k+1} \quad (14)$$

where Δx_{i_e} and Δy_{i_e} represent the horizontal and vertical image displacement respectively. Similarly, the measurement vector, $\mathbf{z}_{k+1}^{(h)}$, belonging to the Kalman filter that handles the head movement is defined as follows,

$$\mathbf{z}_{k+1}^{(h)} = [\Delta x_{i_h} \quad \Delta y_{i_h}]^T_{k+1} \quad (15)$$

where Δx_{i_h} and Δy_{i_h} denote the horizontal and vertical head displacement inside the image space.

Measurement Matrix: The measurement matrix defines the relationship that maps the true state space onto the measurement. In our work, the values that populate the measurement matrices can be derived from Equations (4) and (9), such that these matrices map the screen coordinates onto the image coordinates. The measurement matrix corresponding to the Kalman filter that handles the iris displacement is defined as,

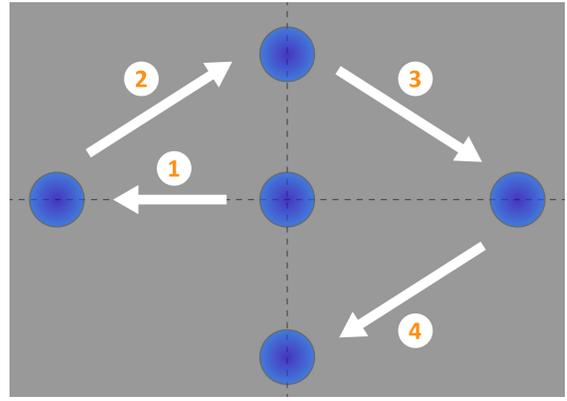


Figure 2. Five visual stimuli were displayed in succession on the monitor screen during a brief calibration procedure in order to collect image-screen coordinate pairs.

$$H_{k+1}^{(e)} = \begin{bmatrix} \frac{(x_{i_e}^{(2)} - x_{i_e}^{(1)})}{(x_s^{(2)} - x_s^{(1)})} & 0 \\ 0 & \frac{(y_{i_e}^{(2)} - y_{i_e}^{(1)})}{(y_s^{(2)} - y_s^{(1)})} \end{bmatrix} \quad (16)$$

while the measurement matrix attributed to the Kalman filter that handles the head movement is defined as,

$$H_{k+1}^{(h)} = \begin{bmatrix} \frac{S_x(x_{i_e}^{(2)} - x_{i_e}^{(1)})}{(x_s^{(2)} - x_s^{(1)})} & 0 \\ 0 & \frac{S_y(y_{i_e}^{(2)} - y_{i_e}^{(1)})}{(y_s^{(2)} - y_s^{(1)})} \end{bmatrix} \quad (17)$$

Measurement Noise and Process Noise: The measurement noise is represented by vector, $\mathbf{v}_{k+1} = [v_{k+1}^x \quad v_{k+1}^y]$, characterized by standard deviations σ_{v_x} and σ_{v_y} in the x- and y-directions, respectively, and similarly the process noise is represented by vector, $\mathbf{w}_{k+1} = [w_{k+1}^x \quad w_{k+1}^y]$, characterized by standard deviations σ_{w_x} and σ_{w_y} in the respective x- and y-directions. The process noise corresponding to the iris movement is taken to represent the characteristics inherent to the visual system itself, such that the standard deviations $\sigma_{w_x}^{(e)}$ and $\sigma_{w_y}^{(e)}$ are therefore set to a low value to model the small, microsaccadic movements performed by the eye during periods of fixation. Similarly, the standard deviations $\sigma_{w_x}^{(h)}$ and $\sigma_{w_y}^{(h)}$ characterizing the process noise corresponding to the head movement are also set to a low value. Values for the standard deviations, $\sigma_{v_x}^{(e)}$ and $\sigma_{v_y}^{(e)}$, corresponding to the iris displacement and standard deviations, $\sigma_{v_x}^{(h)}$ and $\sigma_{v_y}^{(h)}$, corresponding to the head movement that adequately smooth the mouse cursor trajectory after the estimation of noisy image measurements were found experimentally.

Separate Kalman filters are assigned to every screen quadrant, with each filter being characterized by a different measurement matrix corresponding to the screen quadrant for which it is responsible. During tracking, all Kalman filters are updated online to produce an estimate of the POR following

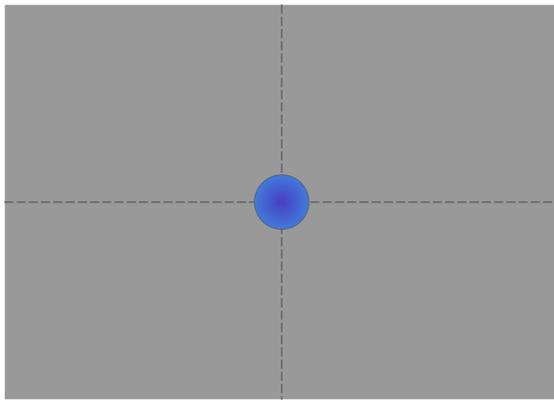


Figure 3. An additional calibration target was displayed at the center of the monitor screen while the participants performed horizontal and vertical head rotations, maintaining a fixed gaze point.

the estimation of the iris center coordinates, such that the on-screen position of the mouse cursor can subsequently be updated according to the Kalman filter estimate that corresponds to the quadrant of interest. In updating the Kalman filters at every time step, we ensure a smooth hand over between one filter and another as the mouse cursor trajectory crosses over adjacent screen quadrants.

III. EXPERIMENTAL RESULTS AND DISCUSSION

To evaluate the proposed eye-gaze tracking method, a group of nine female and male participants having a mean age of 35.1 and standard deviation of 12.7, were recruited for an experimental session. All participants were proficient computer users without any prior experience in the field of eye-gaze tracking, except for one participant who was already accustomed to the technology. The experimental procedure was carried out on a 15.6" notebook display while each participant was seated inside a well-lit indoor environment at an approximate distance of 60 cm from the monitor screen and the camera. At this distance, combined eyeball and head rotations were carried out within a $\pm 15^\circ$ range, corresponding to the width and height of the monitor screen. Image data was acquired by the webcam that was readily available on-board the notebook computer.

Following detection and tracking of the eye region, and iris center localization, each participant was requested to sit through a brief calibration procedure that served to estimate the mapping functions required to transform the iris center and head displacement inside the image space into a POR on the monitor screen. During the calibration procedure, the participants were first instructed to fixate at five visual stimuli appearing in succession on the monitor screen as shown in Figure 2, and requested to maintain a stationary head pose while pairs of image-screen coordinates were collected. Subsequently, an additional calibration target was displayed at the center of the monitor screen as shown in Figure 3, during which the participants were instructed to perform horizontal and vertical head rotations while maintaining their gaze fixed onto the visual target. The five visual stimuli displayed successively during calibration were positioned strategically in order

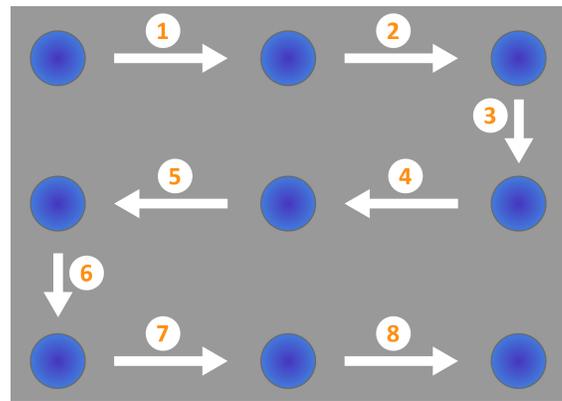


Figure 4. A validation session consisting of nine visual stimuli displayed in succession served to calculate the error of the estimated PORs.

to divide the screen into four separate quadrants, as illustrated in Figure 1. A different mapping function was estimated for each quadrant according to the relationship between the image and screen coordinates collected earlier.

Every participant was then requested to sit through two validation procedures that served to calculate the error between the estimated POR and ground truth data. The first validation procedure consisted of nine visual stimuli which were evenly spread throughout the monitor screen and displayed in succession as shown in Figure 4. The participants were allowed natural head movement, and instructed to move the mouse cursor as close to each visual stimulus as possible and hold its position for a brief period of time such that the on-screen coordinates of the mouse cursor were recorded, as shown in Figure 7 for participant 7. Tables I and II display the mean absolute error (MAE) and standard deviation (SD) in pixels for each validation target in the x- and y-directions. The second validation procedure served to evaluate the accuracy of head pose compensation for a fixed POR on the monitor screen. To this end, the participants were requested to maintain their gaze fixed upon a single visual stimulus displayed on the monitor screen and perform arbitrary head rotations in the horizontal and vertical directions while the mouse cursor coordinates were recorded. Table IV presents the MAE and SD in pixels calculated between the cursor position and validation target in the x- and y-directions. Qualitative results of the iris segmentation method are presented in Figure 6. The proposed method was coded in C using the OpenCV library [31] and was capable of executing in real-time at 18 fps on an Intel Core i5 notebook computer, at 2.60 GHz.

An analysis of the results in Tables I and II reveals that for the majority of the validation targets, the MAE and SD values in pixel units for the x-coordinates of the estimated PORs exceed the error values corresponding to the y-coordinates across all participants. One of the main sources for this consistent discrepancy in error was found to relate to inaccuracies in the estimation of the iris center coordinates, as indicated by the iris segmentation results in Figures 6(a-e). As shown in the binary images, dark colored pixels belonging to the eyelashes are often erroneously included with the iris region

TABLE I. MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION (SD) IN PIXELS, OF THE ESTIMATED ON-SCREEN POR X-COORDINATES FOR EACH VALIDATION TARGET.

Participant	Validation Target								
	1	2	3	4	5	6	7	8	9
	x-Coordinate (MAE (pix), SD (pix))								
1	(18.32, 13.09)	(37.96, 28.02)	(20.81, 11.74)	(28.12, 18.30)	(6.19, 3.05)	(34.36, 5.27)	(9.03, 6.74)	(10.88, 5.08)	(34.09, 13.42)
2	(27.12, 15.40)	(12.91, 13.36)	(38.92, 5.41)	(8.66, 5.46)	(9.03, 8.06)	(47.25, 21.30)	(6.99, 6.00)	(18.11, 12.28)	(37.96, 13.04)
3	(22.00, 13.13)	(17.91, 9.14)	(20.06, 11.27)	(9.95, 7.75)	(10.66, 8.37)	(22.47, 8.10)	(7.67, 5.03)	(20.65, 8.01)	(33.64, 9.38)
4	(20.38, 8.80)	(37.17, 24.06)	(37.61, 6.18)	(32.55, 20.05)	(55.94, 38.20)	(30.23, 0.67)	(12.47, 3.70)	(26.58, 5.93)	(34.91, 7.94)
5	(10.66, 2.14)	(27.80, 14.63)	(58.44, 25.70)	(20.68, 13.11)	(13.64, 3.18)	(38.37, 4.46)	(6.98, 6.23)	(33.85, 18.78)	(36.80, 15.25)
6	(7.72, 4.46)	(20.16, 3.37)	(23.90, 6.73)	(4.16, 2.28)	(25.54, 7.13)	(27.10, 8.28)	(10.81, 12.31)	(15.61, 3.53)	(20.73, 8.51)
7	(29.85, 7.43)	(32.49, 7.69)	(40.82, 14.18)	(11.53, 6.15)	(13.95, 3.89)	(49.05, 10.89)	(13.57, 7.82)	(9.22, 6.09)	(19.35, 4.86)
8	(17.67, 4.95)	(8.07, 0.96)	(17.22, 4.36)	(30.95, 11.36)	(15.37, 15.08)	(23.41, 5.55)	(6.01, 2.38)	(6.10, 3.96)	(27.41, 4.34)
9	(14.22, 10.92)	(17.90, 9.79)	(25.75, 15.29)	(20.49, 17.14)	(17.32, 9.49)	(26.92, 23.10)	(18.88, 8.10)	(10.71, 5.05)	(47.01, 22.97)
Mean	(18.66, 8.92)	(23.60, 12.34)	(31.50, 11.21)	(18.57, 11.29)	(18.63, 10.72)	(33.24, 9.74)	(10.27, 6.48)	(16.86, 7.63)	(32.43, 11.08)

TABLE II. MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION (SD) IN PIXELS, OF THE ESTIMATED ON-SCREEN POR Y-COORDINATES FOR EACH VALIDATION TARGET.

Participant	Validation Target								
	1	2	3	4	5	6	7	8	9
	y-Coordinate (MAE (pix), SD (pix))								
1	(17.58, 5.76)	(17.06, 7.60)	(11.75, 2.27)	(26.60, 20.07)	(16.26, 9.50)	(10.80, 7.35)	(31.84, 16.23)	(6.63, 6.80)	(7.29, 6.97)
2	(11.48, 8.46)	(49.17, 14.13)	(7.75, 2.02)	(13.42, 8.65)	(9.96, 6.17)	(33.21, 19.31)	(22.28, 5.19)	(23.80, 13.06)	(14.37, 8.63)
3	(16.31, 13.45)	(11.97, 9.08)	(16.03, 7.98)	(5.64, 3.65)	(13.16, 11.03)	(13.38, 8.06)	(20.53, 7.51)	(12.20, 7.84)	(10.36, 5.93)
4	(20.03, 8.98)	(30.14, 11.08)	(34.84, 8.96)	(29.59, 7.31)	(19.27, 8.11)	(10.69, 5.13)	(13.87, 6.49)	(18.14, 9.72)	(22.98, 2.82)
5	(4.53, 2.87)	(7.50, 2.91)	(15.12, 7.08)	(12.06, 5.85)	(22.44, 7.50)	(16.74, 1.55)	(44.73, 28.11)	(31.64, 16.13)	(21.39, 18.39)
6	(8.58, 4.55)	(11.50, 4.93)	(29.49, 21.50)	(3.67, 2.43)	(6.34, 3.06)	(20.62, 14.87)	(31.06, 19.01)	(14.28, 10.16)	(11.27, 7.39)
7	(4.58, 3.52)	(26.90, 2.01)	(27.49, 3.28)	(21.19, 5.79)	(4.06, 3.06)	(3.87, 2.03)	(6.97, 4.39)	(22.06, 4.14)	(15.09, 7.48)
8	(56.07, 5.24)	(31.06, 1.83)	(16.35, 2.11)	(35.15, 4.03)	(11.01, 9.04)	(7.63, 4.49)	(23.50, 3.04)	(8.12, 5.36)	(11.75, 4.61)
9	(20.52, 13.49)	(3.54, 3.39)	(10.14, 6.11)	(12.34, 9.41)	(13.60, 6.50)	(37.78, 33.79)	(23.47, 16.18)	(14.08, 7.00)	(11.11, 11.15)
Mean	(17.74, 7.37)	(20.98, 6.33)	(18.77, 6.81)	(17.74, 7.47)	(12.90, 7.11)	(17.19, 10.73)	(24.25, 11.79)	(16.77, 8.91)	(13.96, 8.15)

during segmentation due to their close resemblance in color to dark brown irises. This erroneous inclusion of pixels on either side of the iris region serves to shift the center of mass of the segmented blob of pixels horizontally towards the inner or outer eye corners, away from the true iris center. The horizontal displacement of the iris center gives rise to errors in the estimated PORs, and it was found that even a seemingly trivial error of a few pixels in the estimation of the iris center coordinates inside the image frame could produce a significant POR error at an approximate distance of 60 cm from the monitor screen.

While the majority of the validation targets displayed a higher error in the horizontal component of the estimated PORs, the results for validation target 7 clearly indicate the contrary, that is higher MAE and SD values for the y-coordinates of most participants. The discrepancy in error for this particular validation target relates to its lower left on-screen position as shown in Figure 4, requiring the user to gaze downwards and towards the outer edge of the monitor screen. At this instance, the tracked iris region displaces towards the outer eye corner and becomes partially occluded underneath the upper and lower eyelids, as shown in Figures 6(f-j). The reduced visibility of the true shape of the iris region shifts the localized iris centre downwards, reducing the accuracy of the corresponding POR. Another source of error that reduces the accuracy of the POR y-coordinates in general is the less than ideal positioning of the webcam at the top of the monitor screen, in relation to the positioning of the eyes as the user sits in front of the display. Indeed, commercial systems usually place the tracking device below the monitor screen in order to capture a better view of the visible portion of the eyeball that is not concealed below the eyelid. Being situated at the

TABLE III. MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION (SD) OF THE ERROR IN PIXELS AND VISUAL ANGLE, OF THE ESTIMATED ON-SCREEN POR COORDINATES FOR THE FIRST VALIDATION TARGET.

Participant Number	x	y	x	y
	(MAE (pix), SD (pix))	(MAE (pix), SD (pix))	(MAE (°), SD (°))	(MAE (°), SD (°))
1	(22.20, 11.63)	(16.20, 9.17)	(0.53, 0.28)	(0.39, 0.22)
2	(22.99, 11.15)	(20.60, 9.51)	(0.55, 0.27)	(0.49, 0.23)
3	(18.34, 8.91)	(13.29, 8.28)	(0.44, 0.21)	(0.32, 0.20)
4	(31.98, 12.84)	(22.17, 7.62)	(0.76, 0.31)	(0.53, 0.18)
5	(27.47, 11.50)	(19.57, 10.04)	(0.65, 0.27)	(0.47, 0.24)
6	(17.30, 6.29)	(15.20, 9.77)	(0.41, 0.15)	(0.36, 0.23)
7	(24.43, 7.67)	(14.69, 3.97)	(0.58, 0.18)	(0.35, 0.09)
8	(16.91, 5.88)	(22.29, 4.42)	(0.40, 0.14)	(0.53, 0.11)
9	(22.13, 13.54)	(16.29, 11.89)	(0.53, 0.32)	(0.39, 0.28)
Mean	(22.64, 9.93)	(17.81, 8.30)	(0.54, 0.24)	(0.42, 0.20)

top of the screen, the webcam that is utilized in our work captures a smaller portion of the iris especially when the user gazes downwards, partially occluding the iris region below the eyelid and potentially introducing an error in the estimation of the iris center coordinates. It is, nonetheless, worth noting that since the iris region is localized by its photometric appearance rather than the shape, the proposed method for iris region segmentation was equally capable of detecting the iris region and hence follow its displacement under partial occlusion by the eyelids.

In order to put the error values tabulated in Tables I and II into context, the mean MAE and SD values per participant across all validation targets were subsequently calculated and converted to visual angle at a distance of 60 cm between the user and the monitor screen, as presented in Table III. Given that the resolution of the monitor screen is equal to 1366 × 768 pixels, a mean MAE value of (22.64, 17.81) in pixel units constitutes less than 2% and 3% of the screen

resolution in the horizontal and vertical directions, respectively. Furthermore, at a distance of 60 cm away from the monitor screen, a mean error of (22.64, 17.81) pixels was found to correspond to an error of (0.54°, 0.42°) in visual angle. It is also worth noting that in approximating any head movement performed by the user during tracking, we have also obtained an improvement of around 37% and 59% in visual angle corresponding to the horizontal and vertical directions respectively, over our previous result of (1.46°, 0.71°) obtained through our earlier work without head movement compensation [1]. This improvement further substantiates the benefit of handling head movement during tracking rather than constraining the user to a stationary head pose, where small head displacements performed by the user due to fatigue may adversely affect the accuracy of the estimated PORs if the computed mapping function is valid for a single head pose alone. Indeed, plotting the calculated on-screen iris and head displacements performed by participant 7 during the validation procedure in Figure 5, reveals that the users naturally tend to perform combined eyeball and head movement during tracking in order to move the mouse cursor position close to the validation targets. The low quantitative error obtained by the proposed method may also be corroborated qualitatively as shown in Figure 7, displaying a small distance between the estimated PORs on the monitor screen and the ground truth validation targets. If the average on-screen icon is taken to have an average size of 45×45 pixels, the mean error that is achieved through the proposed method can be considered to be within the footprint of the average on-screen icon and therefore applicable to an HCI scenario.

A second validation procedure aimed to evaluate the accuracy of head movement approximation, by requesting the user to perform horizontal and vertical head rotations while maintaining the gaze fixed upon an on-screen validation target. Visually, the mouse cursor was expected to maintain a fixed on-screen position close to the validation target during this experimental procedure. In response to the head rotations performed during this validation stage, the mouse cursor was noticed to displace in the direction of the head movement and promptly return to its starting position in response to a counter-rotation of the eyeball without a change in the user's gaze. This resulted in mouse cursor trajectories as shown in Figure 8, for three horizontal and vertical head rotations performed by participant 7. As shown in Figure 9, the return positions of the mouse cursor corresponding to a total of five different head rotations performed by the participant were close to the ground truth validation target, hence ensuring that a fixed POR was also maintained on the monitor screen. Plotting the calculated on-screen iris and head displacements for these head rotations in Figure 10, together with the resulting mouse cursor coordinates, reveals that at every head rotation the estimated iris and head displacements reach equivalent amplitudes in opposite directions in agreement with Equation (7). The MAE and SD values across all participants, presented in Table IV in pixels and visual angle calculated at a distance of 60 cm from the monitor screen, reveal low error values between the return positions of the mouse cursor and the validation target. These results indicate the effectiveness of the head

TABLE IV. MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION (SD) IN PIXELS AND VISUAL ANGLE, OF THE ESTIMATED ON-SCREEN POR COORDINATES DURING ARBITRARY HEAD ROTATIONS.

Participant Number	x	y	x	y
	(MAE (pix), SD (pix))	(64.02, 19.20)	(MAE (°), SD (°))	(1.53, 0.46)
1	(45.30, 59.53)	(64.02, 19.20)	(1.08, 1.42)	(1.53, 0.46)
2	(36.56, 11.07)	(16.52, 9.70)	(0.87, 0.26)	(0.39, 0.23)
3	(13.27, 7.07)	(24.96, 16.87)	(0.32, 0.17)	(0.60, 0.40)
4	(48.97, 9.66)	(12.28, 4.53)	(1.17, 0.23)	(0.29, 0.11)
5	(16.93, 8.85)	(10.33, 6.91)	(0.40, 0.21)	(0.25, 0.16)
6	(13.06, 4.65)	(21.44, 5.21)	(0.31, 0.11)	(0.51, 0.12)
7	(29.02, 2.46)	(24.13, 3.05)	(0.69, 0.06)	(0.58, 0.07)
8	(42.24, 23.47)	(28.01, 10.23)	(1.01, 0.56)	(0.67, 0.24)
9	(17.37, 6.20)	(23.12, 13.28)	(0.41, 0.15)	(0.55, 0.32)
Mean	(29.19, 14.77)	(24.98, 9.89)	(0.70, 0.35)	(0.60, 0.24)

pose compensation algorithm in maintaining a consistent POR corresponding to a fixed gaze point in the presence of head rotations.

IV. COMPARISON WITH THE STATE-OF-THE-ART

In order to put the results achieved by the proposed method into context, a comparison with the results reported by other state-of-the-art methods [9][15]–[17][21][24][28] has been provided in Table V. Relevant state-of-the-art methods that estimate a POR on a monitor screen under conditions similar to ours have been chosen. These include methods that allow free head movement [9][16][17][21][24][28] and utilize generic imaging hardware, such as webcams [9][15][17][21][24], for image frame acquisition. Table V presents a comparison of gaze estimation results in visual angle, specifying the number of subjects considered during data collection, the number of calibration points required for POR estimation, the range of head rotation angles performed by the participants, and the distance at which the participants sit from the camera and visual targets.

The results in Table V reveal that our method achieves a POR estimation performance that is comparable to [21]¹ or better than the results reported by relevant state-of-the-art methods in [9][15]–[17][21][24][28]. It is also worth noting that our method achieves this performance by employing fewer calibration points than the methods of [15]–[17][21][24][28], in the absence of training prior to POR estimation unlike the method of [16] and without constraining the user to a stationary head pose by the use of a chin-rest unlike the method of [15]. These characteristics allow for a method that addresses the challenges associated with pervasive eye-gaze tracking in less constrained conditions, by estimating a POR on a monitor screen from images acquired by an integrated webcam inside a notebook computer, under natural head movement, following a brief calibration procedure that does not require prolonged user-cooperation.

V. CONCLUSION

In this paper, we have proposed a passive eye-gaze tracking method to estimate the POR on a monitor screen from low-quality image data acquired by an integrated camera inside a notebook computer, while approximating any head rotations performed by the user during tracking. Following eye

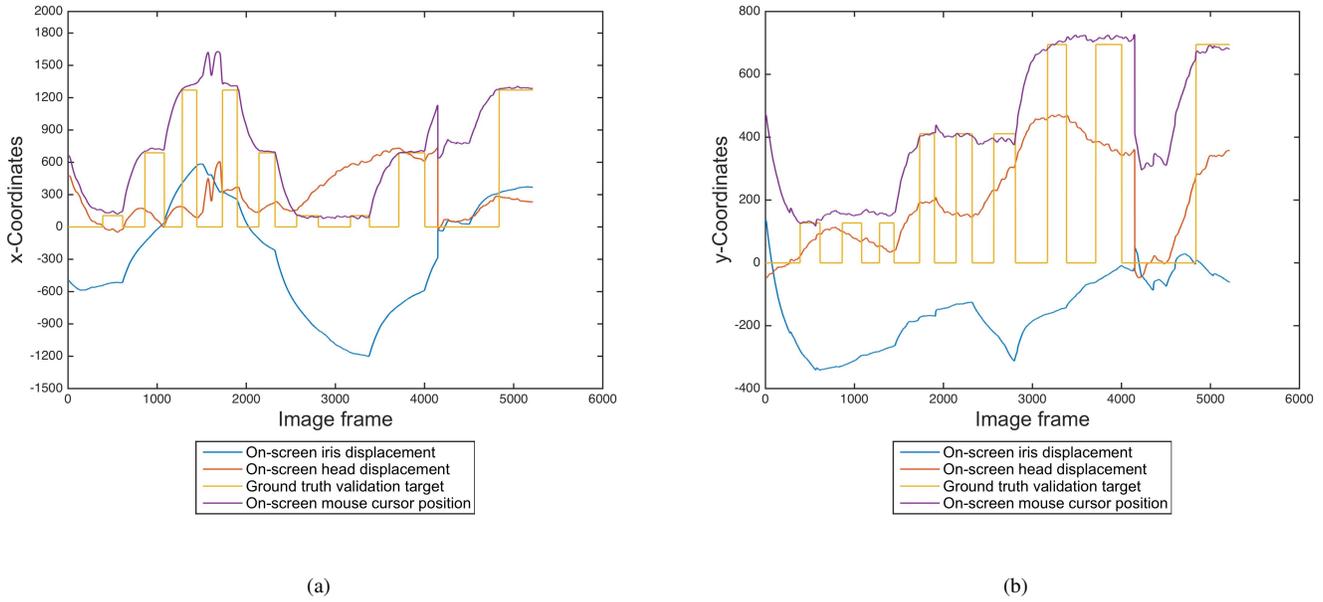


Figure 5. Calculated on-screen iris and head displacements, with respect to reference head and eye poses, during horizontal (a) and vertical (b) head rotations performed by participant 7 throughout the first validation procedure.

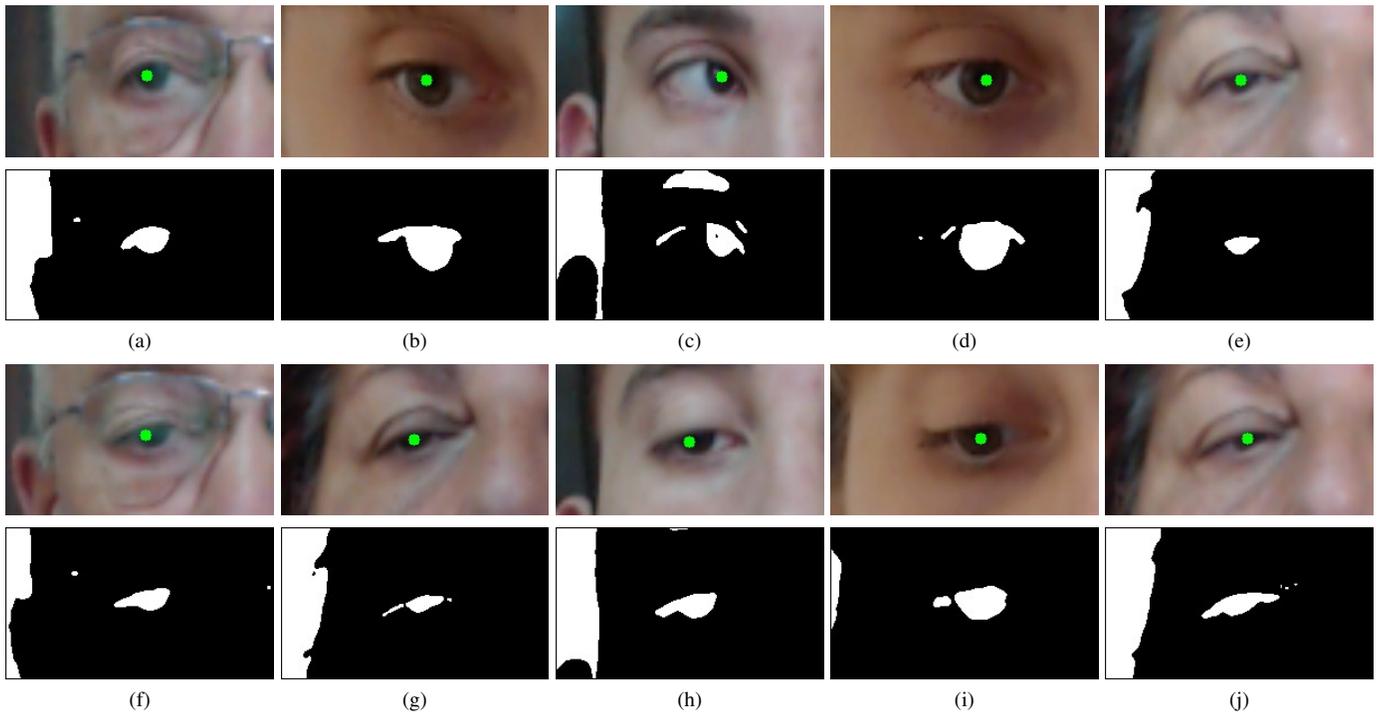


Figure 6. The proposed method for iris region segmentation was capable of localizing the iris center coordinates marked in green, at different eyeball rotations (a-e) and under partial occlusion of the iris region by the eyelids (f-j).

TABLE V. COMPARISON OF THE MEAN ABSOLUTE ERROR (MAE) AND STANDARD DEVIATION (SD) RESULTS OBTAINED BY OUR METHOD WITH RESULTS REPORTED BY RELEVANT STATE-OF-THE-ART METHODS. ^{1,2,3,4} REFER TO RESULTS REPORTED BY THE SAME METHOD, EMPLOYING DIFFERENT CALIBRATION PATTERNS CONTAINING VARYING AMOUNTS OF CALIBRATION POINTS.

Method	Yaw (MAE/°, SD/°)	Pitch (MAE/°, SD/°)	Number of Subjects	Number of Calibration Points	Range of Head Rotation (Yaw/°, Pitch/°, Roll/°)	User-Camera Distance (cm)	User-Target Distance (cm)
Our method	(0.54, 0.24)	(0.42, 0.20)	9	6	(±15, ±15, 0)	60	60
[9]	1.9	2.2	11	5	N/A	75	75
[15]	2.60 (±3.43)	2.61 (±2.45)	5	48	None	75	75
[16]		1.90	4	4000 training samples/user	N/A	N/A	N/A
[17]	3.55 (±0.42)		11	16	N/A	58.42 (±6.98)	58.42 (±6.98)
[21] ¹	0.59 (±0.38)		7	33	N/A	50 - 60	50 - 60
[21] ²	0.63 (±0.39)		7	23	N/A	50 - 60	50 - 60
[21] ³	0.69 (±0.41)		7	18	N/A	50 - 60	50 - 60
[21] ⁴	0.97 (±0.57)		7	9	N/A	50 - 60	50 - 60
[24]		5.06	N/A	12	(≤21.2, ≤9.75, 0)	N/A	N/A
[28]	5.3	7.7	5	Arbitrary head rotations	N/A	220	240

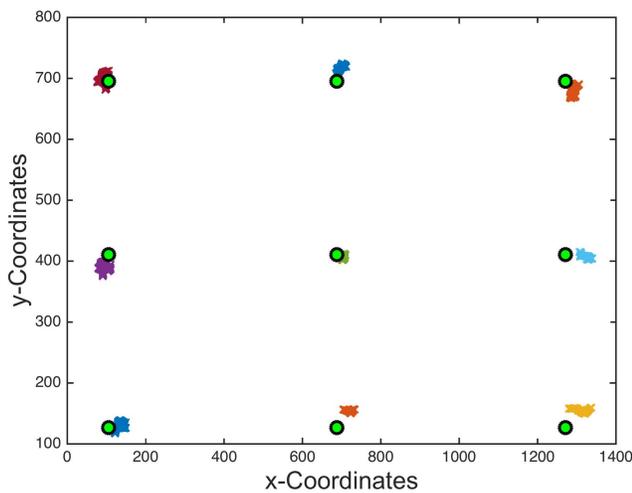


Figure 7. Validation result for participant 7, showing the displayed visual stimuli (green) and the estimated on-screen PORs (colored).

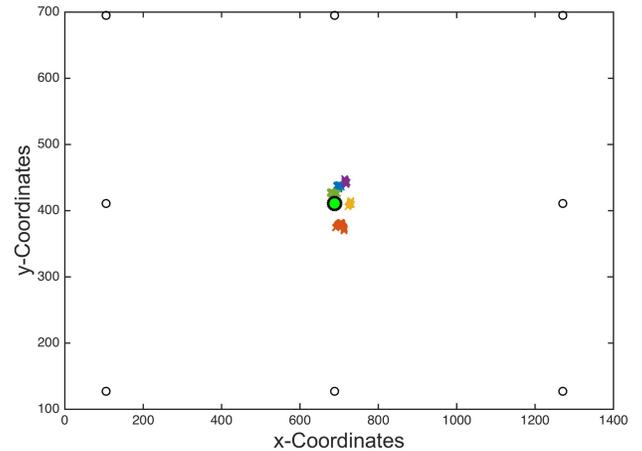


Figure 9. Return positions of the mouse cursor close to the ground truth validation target, corresponding to a total of five different head rotations performed by participant 7.

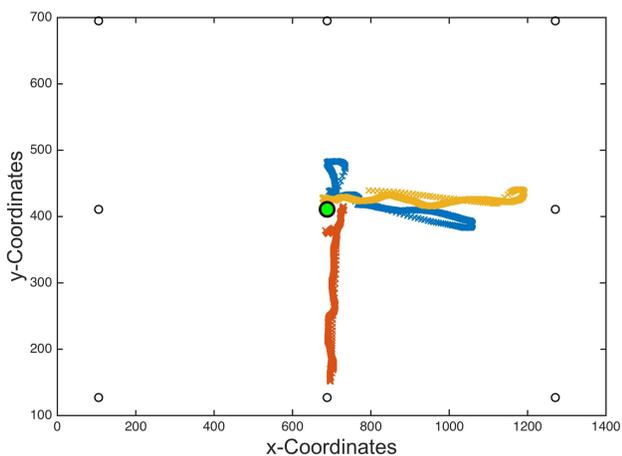


Figure 8. Mouse cursor trajectories in response to three of five head rotations performed by participant 7, keeping a fixed gaze point.

region detection and tracking, we proposed an appearance-based method which allows the localization of the iris center coordinates from low-resolution eye region images. In view of the constraints imposed when working on a computer, approximate linear mapping functions for the iris and head displacement inside the image space were computed during a calibration procedure, in order to map the image information to a POR on the monitor screen. The POR estimates, computed as a function of the iris and head movement, were improved by Kalman filters to smoothen the mouse cursor trajectory.

Two validation procedures were carried out in order to evaluate the proposed POR estimation method with head movement approximation. The experimental results for the first validation procedure revealed a noticeable discrepancy between the error in the x- and y-directions of the estimated PORs, with the error in the x-direction being the dominant between the two. The source of this error was observed to relate to incorrect segmentation of pixels belonging to artifacts, such as the eyelashes, along with the iris region producing a

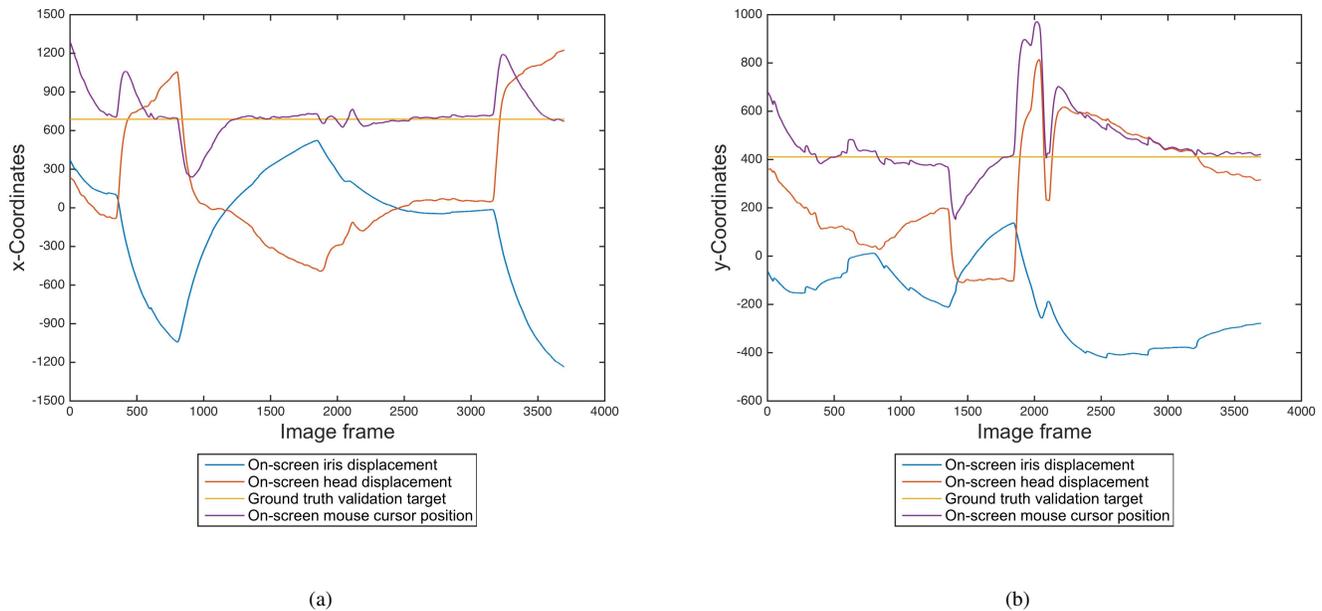


Figure 10. Calculated on-screen iris and head displacements, with respect to reference head and eye poses, having the same amplitude in opposite directions, during horizontal (a) and vertical (b) head rotations maintaining a fixed gaze point.

horizontal shift away from the true iris center. Nonetheless, the proposed method for iris region segmentation was capable of detecting and hence following the displacement of the iris region under partial occlusion by the eyelids, permitting the estimation of the POR in less than ideal conditions. It is noteworthy to mention that the proposed method achieved a low mean MAE of $(0.54^\circ, 0.42^\circ)$, a significant improvement over our previous result of $(1.46^\circ, 0.71^\circ)$ obtained through our earlier work without head movement compensation [1].

The second validation procedure aimed to evaluate the accuracy of head movement approximation. In response to horizontal and vertical head rotations without a change in gaze, the mouse cursor was noticed to displace in the direction of the head movement and promptly return to its starting position, hence retaining a fixed POR on the monitor screen. A low mean MAE of $(0.70^\circ, 0.60^\circ)$ was achieved across all participants, indicating a small distance between the return positions of the mouse cursor and the validation target.

Future work aims to consider the non-linearities associated with eyeball and head rotations, approximated by planar image displacement of the respective iris center and head region in this work, in order to permit a wider range of eyeball and head rotations during tracking.

ACKNOWLEDGMENT

This work forms part of the project *Eye-Communicate* funded by the Malta Council for Science and Technology through the National Research & Innovation Programme (2012) under Research Grant No. R&I-2012-057.

REFERENCES

- [1] S. Cristina and K. P. Camilleri, "Cursor control by point-of-regard estimation for a computer with integrated webcam," in The 8th International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP), 2014, pp. 126–131.
- [2] R. J. K. Jacob, "What you look at is what you get: eye movement-based interaction techniques," in Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, 1990, pp. 11–18.
- [3] J. L. Levine, "An eye-controlled computer," in IBM Thomas J. Watson Research Center Res. Rep. RC-8857, Yorktown Heights, N.Y., 1981.
- [4] S. J. Lee, J. Jo, H. G. Jung, K. R. Park, and J. Kim, "Real-Time Gaze Estimator Based on Driver's Head Orientation for Forward Collision Warning System," IEEE Transactions on Intelligent Transportation Systems, vol. 12, 2011, pp. 254–267.
- [5] M. Shahid, T. Nawaz, and H. A. Habib, "Eye-Gaze and Augmented Reality Framework for Driver Assistance," Life Science Journal, vol. 10, 2013, pp. 1571–1578.
- [6] A. Bulling, A. T. Duchowski, and P. Majaranta, "PETMEI 2011: The 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction," in Proceedings of the 13th International Conference on Ubiquitous Computing: UbiComp 2011, 2011.
- [7] D. W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, 2010, pp. 478–500.
- [8] R. Valenti, A. Lablack, N. Sebe, C. Djeraba, and T. Gevers, "Visual Gaze Estimation by Joint Head and Eye Information," in 20th International Conference on Pattern Recognition, Aug. 2010, pp. 3870–3873.
- [9] R. Valenti, N. Sebe, and T. Gevers, "Combining Head Pose and Eye Location Information for Gaze Estimation," IEEE Transactions on Image Processing, vol. 21, 2012, pp. 802–815.
- [10] W. Haiyuan, Y. Kitagawaa, and T. Wada, "Tracking Iris Contour with a 3D Eye-Model for Gaze Estimation," in Proceedings of the 8th Asian Conference on Computer Vision, Nov. 2007, pp. 688–697.

- [11] T. Moriyama, T. Kanade, J. Xiao, and J. F. Cohn, "Meticulously Detailed Eye Region Model and Its Application to Analysis of Facial Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, 2006, pp. 738–752.
- [12] F. Timm and E. Barth, "Accurate Eye Centre Localisation by Means of Gradients," in *Proceedings of the Sixth International Conference on Computer Vision Theory and Applications*, Mar. 2011, pp. 125–130.
- [13] E. G. Dehkordi, M. Mahlouji, and H. E. Komleh, "Human Eye Tracking Using Particle Filters," *International Journal of Computer Science Issues*, vol. 10, 2013, pp. 107–115.
- [14] J. Mansanet, A. Albiol, R. Paredes, J. M. Mossi, and A. Albiol, "Estimating Point of Regard with a Consumer Camera at a Distance," *Pattern Recognition and Image Analysis*, 2013, pp. 881–888.
- [15] W. Sewell and O. Komogortsev, "Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network," in *Proceedings of the 28th International Conference Extended Abstracts on Human Factors in Computing Systems*, 2010, pp. 3739–3744.
- [16] R. Stiefelwagen, J. Yang, and A. Waibel, "Tracking Eyes and Monitoring Eye Gaze," in *Proceedings of the Workshop on Perceptual User Interfaces*, Oct. 1997, pp. 98–100.
- [17] C. Holland and O. Komogortsev, "Eye Tracking on Unmodified Common Tablets: Challenges and Solutions," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, Mar. 2012, pp. 277–280.
- [18] K. Kunze, S. Ishimaru, Y. Utsumi, and K. Kise, "My Reading Life - Towards Utilizing Eyetracking on Unmodified Tablets and Phones," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing*, 2013, pp. 283–286.
- [19] W. Z. Zhang, Z. C. Wang, J. K. Xu, and X. Y. Cong, "A method of gaze direction estimation considering head posture," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 2, 2013, pp. 103–111.
- [20] E. Demjen, V. Abosi, and Z. Tomori, "Eye tracking using artificial neural networks for human computer interaction," *Physiological Research*, vol. 60, 2011, pp. 841–844.
- [21] T. Okabe, F. Lu, Y. Sugano, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, 2014, pp. 2033–2046.
- [22] K. Kim and R. S. Ramakrishna, "Vision-based eye-gaze tracking for human computer interface," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, 1999, pp. 324–329.
- [23] N. M. Piratla and A. P. Jayasumana, "A neural network based real-time gaze tracker," *Journal of Network and Computer Applications*, 2002, pp. 179–196.
- [24] H. Wang, C. Pan, and C. Chaillou, "Tracking eye gaze under coordinated head rotations with an ordinary camera," in *9th Asian Conference on Computer Vision*, vol. 5995, 2010, pp. 120–129.
- [25] R. Ronsse, O. White, and P. Lefevre, "Computation of gaze orientation under unrestrained head movements," *Journal of Neuroscience Methods*, vol. 159, no. 1, 2007, pp. 158–169.
- [26] B. L. Nguyen, Y. Chahir, M. Molina, C. Tijus, and F. Jouen, "Eye gaze tracking with free head movements using a single camera," in *Symposium on Information and Communication Technology*, 2010, pp. 108–113.
- [27] R. Dhyawala, J. Dhariwal, and N. Nain, "Eye gazing with low resolution web-cam images using artificial neural network," in *Proceedings of the International Conference on Advances in Computer, Electronics and Electrical Engineering*, 2012, pp. 443–447.
- [28] H. Yamazoe, A. Utsumi, T. Yonezawa, and S. Abe, "Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions," in *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications*, 2008, pp. 245–250.
- [29] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 499–504.
- [30] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jul. 2001, pp. 1231–1238.
- [31] "OpenCV," 2014, URL: <http://http://opencv.org/> [accessed: 2014-04-21].
- [32] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," in *Proceedings of the GraphiCon*, 2003, pp. 85–92.
- [33] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, 1960, pp. 35–45.

A Framework for Ensuring Non-Duplication of Features in Software Product Lines

Amal Khtira¹, Anissa Benlarabi², Bouchra El Asri³
 IMS Team, SIME Laboratory, ENSIAS, Mohammed V University
 Rabat, Morocco

¹amalkhtira@gmail.com, ²a.benlarabi@gmail.com, ³elasri@ensias.ma

Abstract—Since the emergence of Software Product Line Engineering, the requirements evolution issue has been addressed by many researchers and many approaches have been proposed. However, most studies focused on evolution in domain engineering while application engineering has not received the same attention. During the evolution of a derived product, new features are added or modified in the application model, which may cause many model defects, such as inconsistency and duplication. These defects are introduced to the existing models from the non-verified specifications related to the SPL evolutions. Since these specifications are most of the time expressed in natural language, the task of detecting defects becomes more complicated and error-prone. The aim of this paper is to present a framework that transforms both the SPL feature models and the specification of a new evolution into a more formal representation and provides algorithms to determine the duplicated features between the specification and the existing models. In addition, we describe a support tool created based on the framework and we evaluate the efficacy of our approach using an open source product line.

Keywords—Software Product Line Evolution; Domain Engineering; Application Engineering; Feature Duplication; Natural Language Processing.

I. INTRODUCTION

Feature duplication, as described in [1], occurs when two or more features of the same semantics exist in a feature model of a SPL. SPLs, contrarily to single software, have emerged as a solution to develop different applications based on a core platform. The adoption of SPLs by companies enables them to reduce time to market, to reduce cost and to produce high quality applications. Another major advantage of SPLs is the reuse of core assets to generate specific applications according to the need of customers.

The Software Product Line Engineering (SPLE) approach consists of two processes, namely, domain engineering and application engineering [2]. During these processes, a number of artefacts are produced which encompass requirements, architecture, components and tests. Domain engineering involves identifying the common and distinct features of all the product line members, creating the design of the system and implementing the reusable components. During application engineering, individual products are derived based on the artefacts of the first process, using some techniques of derivation.

Many issues related to SPLE have been addressed both by researchers and practitioners, such as reusability, product derivation, variability management, etc. The focus of our study will be on SPL evolution. Evolution is defined by Madhavji et al. [3] as “a process of progressive change and cyclic adaptation over time in terms of the attributes, behavioral

properties and relational configuration of some material, abstract, natural or artificial entity or system”. This definition applies to different domains, including software engineering.

In the literature, several studies have dealt with evolution in Software Product Lines (SPLs). Xue et al. [4] presented a method to detect changes that occurred to product features in a family of product variants. In order to support agile SPL evolution, Urli et al. [5] introduces the Composite Feature Model (CFM), which consists of creating small Feature Models (FMs) that corresponds each to a precise domain. Other approaches, such as Ahmad et al.’s [6], focused on the extraction of architecture knowledge in order to assess the evolutionary capabilities of a system and to estimate the cost of evolution. Some papers focused on the co-evolution of different elements of SPLs [7].

Based on the literature, we have found that most of the studies addressing software evolution focus on domain engineering, while application engineering has not received the same interest. However, the experience has proven in many industrial contexts that systems continue to change even after the product derivation. This change can be the source of many problems in the product line such as inconsistency and duplication. Indeed, the core assets of the product line and the artefacts of derived products are most of the time maintained by different teams. Moreover, developers under time pressure can forget to refer to the domain model before starting to implement the changes. For these reasons and others, duplication in SPL can easily happen. Since the requirements related to SPL evolutions are most of the time expressed in the form of natural language specifications, the task of model verification becomes difficult and error-prone. In order to simplify the detection of defects, many studies have proposed methods to transform natural language specifications into formal or semi-formal specifications [8][9][10].

In this paper, we describe our framework that consists of unifying the representation of the natural language specifications and the existing domain and application models, then detecting duplicate features using a set of algorithms. We extend the work presented in [1] by:

- Completing the background of our work by adding an overview of the presentation of specifications related to SPL evolutions.
- Presenting an improved version of the proposed framework and describing its different processes in details.
- Enhancing the formalization of the framework concepts.

- Providing the pseudo-code of the algorithm of duplication detection.
- Describing the architecture and main functionality of a support tool created to evaluate the framework.
- Applying the proposed solution to an open source case study.

The remainder of the paper is structured as follows. Section II gives an overview of the background of our study and describes the problem we are dealing with. In Section III, we present the basic concepts and the overview of the proposed framework. In Section IV, we provide a formalization of the basic concepts before describing the algorithm of deduplication. Section V presents the architecture and the main features of the automated tool created based on the framework. An application of the framework on a case study is presented in Section VI. Section VII positions our approach with related works. The paper is concluded in Section VIII.

II. BACKGROUND AND OBJECTIVE

In this section, we introduce the background of our study. First, we present the SPLE paradigm. Then, we highlight the problem of duplication when evolving products in application engineering, and finally, we give an insight of the presentation of specifications related to SPLs, before explaining the objective of our work.

A. Software Product Line Engineering

A SPL is defined by Clements and Northop [11] as “*a set of intensive-software systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way*”. The main goals of a SPL are to reduce the cost of developing software products, to enhance quality and to promote reusability.

The domain engineering phase of the SPLE framework is responsible for defining the commonality and variability of the applications of the product line. Capturing the common features of all the applications increases the reusability of the system, and determining the variant features allows the production of a large number of specific applications that satisfy different needs of customers. When the domain model is ready, the next step consists of creating the design of the system that contains the software components and their relationships. Those components are then implemented and the code of the product line is generated.

The process of creating a specific product based on a SPL is referred to as product derivation or product instantiation. Product derivation consists of taking a snapshot of the product line by binding variability already defined in the domain engineering and using it as a starting point to develop an individual product. This process is applied during application engineering phase and is responsible for instantiating all the artefacts of the product line, i.e., model, design, components, etc.

In order to document and model variability in SPL, many approaches have been proposed. For instance, Pohl et al. [2] proposed the orthogonal variability model to define the

variability in a dedicated model, while other papers preferred to integrate the variability in the existing models, such as UML models or FODA models [12]. Another approach proposed by Salinesi et al. [13] used a constraint-based product line language. In our approach, we will use the FODA method.

B. Duplication of Features during SPL Evolution

The goal of SPLE is to make an up-front investment to create the platform. Indeed, during domain engineering, the requirements of all the potential applications are captured, and as far as possible, the scenarios of the possible changes have to be predicted and anticipated. The evolution and maintenance of the product line are conducted through several iterations until the platform becomes as stable as possible. As new evolutions arise, the domain artefacts are adapted and refined.

On the one hand, the team responsible for developing and maintaining the product line studies the requirements of each customer and derives specific applications that respond to these requirements. On the other hand, a different team takes in charge the maintenance of each application. Following the logic of SPLE, the derived applications are not supposed to change much, but the experience has shown that this assumption is not always true. In fact, even after the derivation of a specific product, new demands can be received from the customer, either changes to existing features or addition of new ones.

During the maintenance of a product, duplication of knowledge can easily happen when evolving the model, the design or the code. In [14], four categories of duplication are distinguished:

- **Imposed duplication:** Developers cannot avoid duplication because the technology or the environment seems to impose it.
- **Inadvertent duplication:** This type of duplication comes about as a result of mistakes in the design. In this case, the developers are not aware of the duplication.
- **Impatient duplication:** When the time is pressing and deadlines are looming, developers get impatient and tend to take shortcuts by implementing as quick as possible the requirements of customers. In these conditions, duplication is very likely to happen.
- **Inter-developer duplication:** Different people working on one product can easily duplicate information.

In the context of SPLE, at least the three last categories might occur. Indeed, when a derived application is shipped, developers responsible for maintaining it do not have a clear visibility of the domain model because another team conceived it. Thus, developers of the application may add features which are already satisfied in the domain model and have only to be derived or configured. In addition, under time pressure, developers do not refer to the application model and might add features which are already implemented.

C. Specifications of SPL Evolutions

In software engineering projects, specifications are an important input of the requirements analysis. These specifications

are most of the time written in natural language. Indeed, this communication channel is obviously what customers often use to express easily what they expect from the system and to explain their perception of the problem. However, specifying requirements in a natural language frequently makes them prone to many defects. In [15], Meyer details seven problems with natural language specifications: noise, silence, over-specification, contradictions, ambiguity, forward references and wishful thinking. Lami et al. [16] focused on other defects, namely the ambiguity, the inconsistency and the incompleteness. In order to deal with these problems, several methods have been proposed in the literature to transform natural language specifications to formal or semi-formal specifications [8][9][10].

In the case of software product lines, the domain model of the entire product line and the application models of the derived products can be expressed using feature models, while the specifications of the new evolutions concerning a derived product can be expressed using natural language. In order to avoid the introduction of defects (Duplication in our case) into the existing feature models, we need to formalize the presentation of these specifications, which facilitates their verification against the SPL models.

D. Objective

To the best of our knowledge, few attempts have dealt with feature duplication when evolving derived products of SPLs. Hence, the aim and contribution of this paper is to provide a framework that helps developers avoid duplication in a SPL when evolving a specific product. To this end, we will first unify the presentation of the domain and application feature models of the product line and the natural language specifications of an evolution related to a derived product. Then, an algorithm is proposed to detect the duplicated features between these two inputs.

III. A FRAMEWORK TO AVOID DUPLICATION WHEN EVOLVING DERIVED PRODUCTS

In this section, we first provide a short definition of the basic concepts used in the framework, then we present the overview of the framework.

A. Basic Concepts

Before going any further, we will give an insight of the basic concepts used in the framework.

Requirement: According to [17], a Requirement is:

- 1) A condition or capability needed by a user to solve a problem or achieve an objective.
- 2) A condition or capability that must or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.
- 3) A documented representation of a condition or capability as in (1) or (2).

Domain Model: A domain is a family of related products, and the domain model is the representation of all the different and common features of these products. There are many types

of domain models, but the most interesting are the feature model [12] and the variability model [2].

Application Model: The model corresponding to an individual application. It is generated by binding the variability of the domain model in a way that satisfies the needs of a specific customer [2].

Feature: A feature is the abstraction of functional or non-functional requirements that help characterize the system and must be implemented, tested, delivered, and maintained [18][19].

Feature Model: It is the description of all the possible features of a software product line and the relationships between them. The most common representation of feature models is by using the FODA feature diagram proposed by [12]. The feature diagram is a tree-like structure where a feature is represented by a node and sub-features by children nodes. In basic feature models, there are four kinds of relationships between features:

- **Mandatory:** The relationship between a child feature and a parent feature is mandatory when the child is included in all the products in which its parent exists.
- **Optional:** The relationship between a child feature and a parent feature is optional when the child feature may or may not be present in a product when its parent feature is selected.
- **Alternative:** The relationship between a set of child features and a parent feature is alternative when only one feature of the children can be included in a product where its parent is present.
- **Or:** The relationship between a set of child features and a parent feature is an or-relationship when one or more of them can be selected in the products in which its parent feature exists.

Variation Point: Variation points are places in a design or implementation that identify the locations at which variation occurs [20].

Variant: It is a single option of a variation point and is related to the latter using a variability dependency [21].

Specification: Requirements specification is a description of the intended behavior of a software product. It contains the details of all the features that have to be implemented during an evolution of the system.

B. The Framework in a Nutshell

With the large number of features in the SPLs, the manual checking of duplication becomes a complicated and an error-prone task. In order to deal efficiently with the problem of duplication during the evolution of derived products, we proposed the framework depicted in Figure 1 as an attempt to set an automated deduplication tool. As stated in [22], the objective of this framework is to transform the new specifications and the existing feature models into a formal representation, which facilitates the detection of duplication between these two inputs.

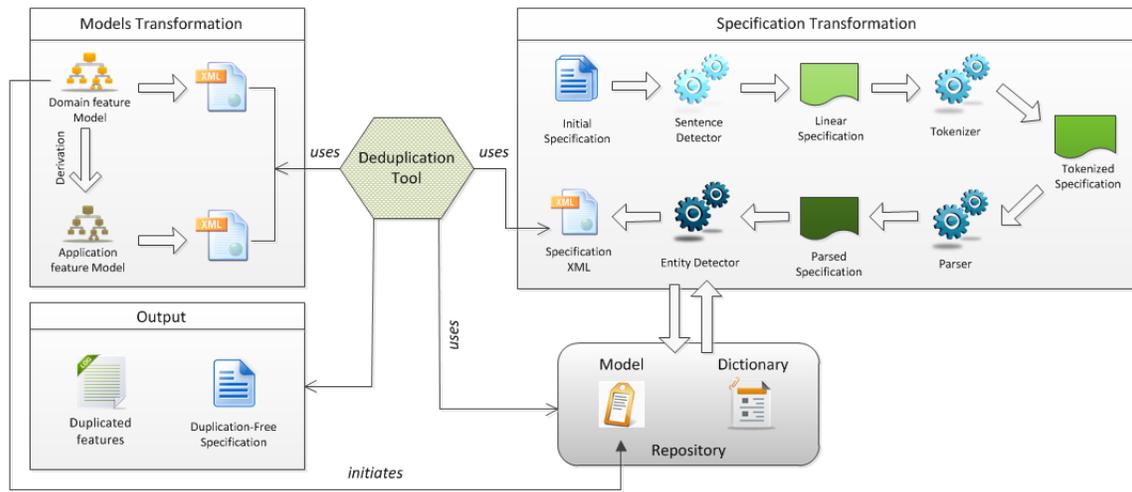


Figure 1. The Overview of the Framework.

1) *Models Transformation*: Feature-oriented software development (FOSD) [23] is a development paradigm based on the FODA method. This paradigm can be used to develop product lines and to do domain engineering. When other methods lack clean mapping between features, architecture and implementation artefacts, FOSD aims at generating automatically the software products. Hence, tools like GUIDSL [24] and FeatureIDE [25] have been proposed. These tools present features in an interactive form and allow a flexible selection of the features of a derived product.

During this step of the framework, we use the FeatureIDE tool in order to provide a formal presentation of the domain model of the product line and the application model of the derived product. FeatureIDE [25] is an open source framework for software product line engineering based on FOSD. This framework supports the entire life-cycle of a product line, especially domain analysis and feature modeling. Indeed, the framework provides the possibility of presenting graphically the feature model tree and generates automatically the corresponding XML source.

2) *Specification Transformation*: This process of the framework is responsible for transforming the natural language specifications of a derived product to an XML document. For this, we use the OpenNLP library [26], which is a machine learning based toolkit for the processing of natural language text. The remainder of this subsection details the different steps and artefacts involved in this process. In [27], we described the different steps of parsing:

Initial Specification: The main input of this process is the specification of a new evolution related to a derived product. The specification contains the features that have to be implemented in this specific product, expressed in natural language. In the context of our framework, we consider that a feature is the association of a variation point and a variant.

Sentence Detector: The first step of the process consists of detecting the punctuation characters that indicate the end of sentences. After the detection of all sentence boundaries, each sentence is written in a separated line. The output of this

operation is a new specification that contains a sentence per line.

Tokenizer: This step consists of dividing the resulted sentences of the previous step into tokens. A token can be a word, a punctuation, a number, etc. At the end of this action, all the tokens of the specification are separated using whitespace characters, such as a space or line break, or by punctuation characters.

Parser: The parser analyses the sentences of the specification in order to determine the roles of the different tokens based on the rules of the language grammar (e.g., noun, verb, adjective, adverb). In our case, the language used in the specifications is English. A parser marks all the words of a sentence using a POS tagger (Part-Of-Speech tagger) and converts the sentence into a tree that represents the syntactic structure of the sentence. Figure 2 illustrates an example of parsing for a natural language requirement related to the case study presented in Figure 5.

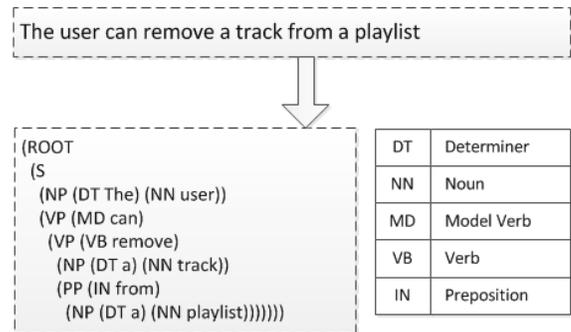


Figure 2. An Example of Syntax Parsing

This action allows us to have an exact understanding of the sentence. For example, it enables us to confirm whether the action of a verb is affirmative or negative, and whether a requirement is mandatory or optional.

Entity Detector: During this step, we detect semantic

entities in the specification. In our study, we are interested in the parts of the sentences considered as variation points and variants. To carry out this task, we use the repository and especially the model where all the domain specifications are tagged.

The example given in Figure 3 represents a sentence processed by the entity detector. The output is a sentence with markup for the detected variant.

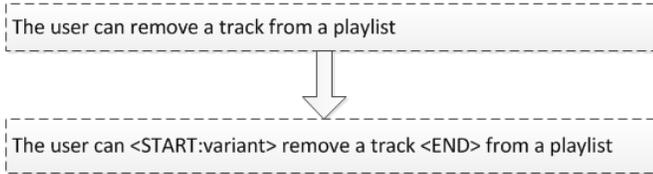


Figure 3. An Example of Detecting Entities

In order to measure the precision of entity recognition, we use the evaluation tool of OpenNLP that calculates the accuracy of the used model.

3) *Repository*: The repository contains two components:

- **The model**: It is initially created based on the domain model of the SPL and contains the different features classified by categories, especially <variation point> and <variant>.
- **The dictionary**: It contains the set of synonyms and alternatives for all the concepts used in the system.

The repository is initially populated based on the domain model of the product line. So that the repository follows the evolution of the product line and its derived products, the new concepts detected in the specification are added to the initial repository.

4) *Deduplication Tool*: This tool contains a set of algorithms of features verification. In this paper, we focus on the algorithm of deduplication between the feature models and the specifications of new evolutions. The other algorithms are responsible for the verification of non duplication in specifications [27] and in feature models. Before describing the algorithm, we need to define some predicates.

Equivalence: We consider that a variation point (resp. a variant) is equivalent to another variation point (resp. variant) if they both implement the same functionality, which means that they have the same semantics. The equivalents of the different variation points and variants of the system are stored in the repository.

Example: The variant "On-line Sales" associated to the variation point "Sales" is equivalent to the new variant "e-sales".

Duplication: We consider that a feature of the specification is duplicated if the associated variation point and variant have equivalents in the application model or the domain model.

The aim of the algorithm is thus to verify the non-duplication of all the features of the initial specification in order to generate a new correct specification. Indeed, for

each feature of the initial specification, the algorithm verifies whether the associated variation point and variant have equivalents in the domain model and the application model. The detection of equivalence is carried out based on the *Repository* content.

5) *Output*: The output of the framework is a duplication-free specification that contains distinct features, and the list of the duplicate features. The two outputs are sent to the user in order to verify his initial requirements and change them in case of need.

IV. AN ALGORITHM FOR DUPLICATION-FREE SPL

In this section, we provide the formalization of the basic concepts used in the framework, then we describe the deduplication algorithm. The inputs of the algorithm, as depicted in Figure 1, are the XML trees of the domain model, the application model and the specification.

A. Formalizing the Basic Concepts

Prior to explaining the algorithm, a certain number of predicates must be defined. We denote by D the domain model. PD is the set of variation points of D , and VD is the set of variants of D .

$$PD = \{pd_1, pd_2, \dots, pd_m\}$$

$$\forall pd_i \in PD \quad \exists VD_i \text{ where } VD_i = \{vd_{ij} \mid j \in \mathbb{N}\}$$

Thus:

$$VD = \bigcup_{i=1}^m VD_i$$

Similarly, we denote by A the application model of a derived application. PA is the set of variation points of A , and VA is the set of variants of A .

$$PA = \{pa_1, pa_2, \dots, pa_m\}$$

$$\forall pa_i \in PA \quad \exists VA_i \text{ where } VA_i = \{va_{ij} \mid j \in \mathbb{N}\}$$

Thus:

$$VA = \bigcup_{i=1}^m VA_i$$

As a reminder:

$$PA \subseteq PD \quad \text{and} \quad VA \subseteq VD$$

Finally, we denote by S the specification of a new evolution. P is the set of variation points of S , and V is the set of variants of S .

$$P = \{p_1, p_2, \dots, p_n\}$$

$$\forall p_i \in P \quad \exists V_i \text{ where } V_i = \{v_{ij} \mid j \in \mathbb{N}\}$$

Thus:

$$V = \bigcup_{i=1}^n V_i$$

It has to be noted that P and V are not subsets of PA and VA .

B. The Algorithms of Duplication Detection

In order to verify whether a feature is duplicated when implementing a new specification, we propose two algorithms [22]. The first algorithm (Algorithm 1) carries out a comparison between the specification and the domain model. In the second algorithm (Algorithm 2), the comparison is between the specification and the application model of a derived product.

Algorithm 1 Detecting duplication between S and D

```

Principal Lookup :
for each  $p_i \in P$  do
  for each  $pd_k \in PD$  do
    if  $Equiv(p_i, pd_k)$  then
      Secondary Lookup :
      for each  $v_{ij} \in V_i$  do
        for each  $vd_{kl} \in VD_k$  do
          if  $Equiv(v_{ij}, vd_{kl})$  then
            NoticeDuplication( $p_i, v_{ij}, pd_k, vd_{kl}$ )
            Continue Secondary Lookup
          end if
        end for
      end for
    end if
  end for
  Continue Principal Lookup
end if
end for
end for

```

Algorithm 2 Detecting duplication between S and A

```

Principal Lookup :
for each  $p_i \in P$  do
  for each  $pa_k \in PA$  do
    if  $Equiv(p_i, pa_k)$  then
      Secondary Lookup :
      for each  $v_{ij} \in V_i$  do
        for each  $va_{kl} \in VA_k$  do
          if  $Equiv(v_{ij}, va_{kl})$  then
            NoticeDuplication( $p_i, v_{ij}, pa_k, va_{kl}$ )
            Continue Secondary Lookup
          end if
        end for
      end for
    end if
  end for
  Continue Principal Lookup
end if
end for
end for

```

The first loop of these algorithms consists of comparing each variation point of the specification with the variation points of the feature model. When an equivalent is found, the algorithms start another comparison between the variants corresponding to the variation point of the specification and the variants related to the equivalent variation point of the feature model. If a variant is detected, the feature corresponding to the pair (variation point, variant) is considered as duplication.

Although the two algorithms are similar, but it is necessary to implement them both because the decision when detecting a duplication will be different. Indeed, if the new feature exists already in the application model, nothing has to be done. However, if the new feature exists in the domain model but not in the application model, a derivation of this feature is necessary.

Result:

For each algorithm, the duplicate features are stored in a log file that will be sent to the customer in order to inform him

of the duplication and its location. When the customer confirms the duplication, the duplicate features are removed from the initial specification and a new duplication-free specification is generated.

V. AUTOMATED TOOL

At the aim of automatizing the proposed framework, a support tool is currently under development [28]. In this section, we describe the implementation of this tool. Then, we present some of the features it provides.

A. Tool Implementation

The tool is a thick-client java-based application built around Eclipse IDE. A major factor of this choice is the adaptability of Eclipse and its large community of plugins compared to its competitors. Figure 4 presents the architecture of the proposed tool.

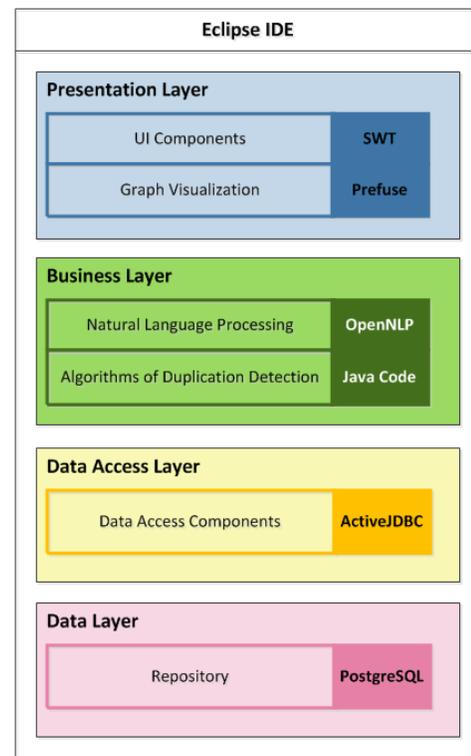


Figure 4. The Tool Architecture.

The tool is composed of four layers:

- Presentation Layer:** This layer enables the communication between the user and the application. The tool interface is created using SWT [29], which is an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented. In our case, the application is built on Windows. So, SWT will use Windows facilities to create the interface. Another aspect managed in this layer is the visualization of the processed specifications as the form of a graph. For this, we use Prefuse [30], which

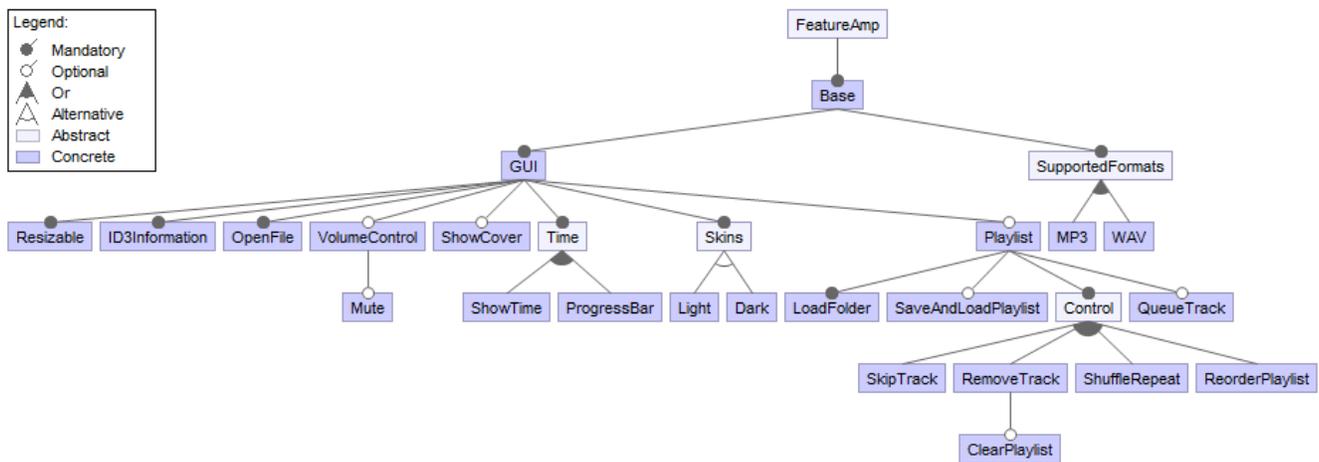


Figure 5. The Domain Feature Model of FeatureAMP.

is an open source toolkit that provides a visualization framework for the Java programming language.

- **Business Layer:** This layer is responsible for the definition of business operations. In our tool, we process textual specifications and we transform them to a tree-like document using OpenNLP. In addition, we implement the algorithms of duplication detection using Java code.
- **Data Access Layer:** This layer provides data to the Business Layer and updates the Data Layer with new information. For this, we use ActiveJDBC [31].
- **Data Layer:** In order to store the content of the repository, we use PostgreSQL [32], which is an open source object-relational database system. In the repository are stored all the domain features, their categories, and their synonyms.

B. Tool Features

The final goal of the automated tool is to detect duplication between new specifications and feature models during SPL evolution. Thus, the main features of the tool are as follows:

- The upload of a textual specification and an XML feature model.
- The transformation of a specification into a tree.
- The detection of duplicate features in a specification.
- The detection of duplicate features in a feature model.
- The comparison of features between a feature model and a specification to detect duplication between them.
- The creation and update of the repository.

The tool provides also some auxiliary features that help achieve the target goal:

- The visualization of the processed specification as a graph.

- The distinction of duplicate features with a different color in the graph.
- The binding of new variants with the corresponding variation points in the repository.
- The manual and automatic update of the repository based on a new specification or a new feature model.
- The re-processing of the specification after a modification of the repository.
- The generation of a log with information about the detected duplicate features.
- The sending of the log to the user via email.

VI. CASE STUDY

In a previous work [28], we evaluated our framework using a CRM SPL. In the feature model of this tool, the features are expressed in the form of sentences. In this paper, we will apply the framework on a different case study which is the FeatureAMP tool [33]. FeatureAMP is an open source audio player product line. The features of this tool are expressed using one word, which makes our task more difficult, because we have to transform the word into a sentence before comparing it to the new requirements.

A. The Feature Models

Figure 5 depicts the domain feature model of FeatureAMP. This tool supports two formats, MP3 and WAV, and provides many functionality such as the play-list management and the volume control.

The main interface of our tool presented in Figure 6 contains two main entries (File) and (Repository). The first entry enables the import of the specification and the feature model. The second entry is used to display the content of the repository.

In order to upload the XML source of the domain model, we use the option "Open Feature Model". The interface presented in Figure 7 displays the uploaded feature model and

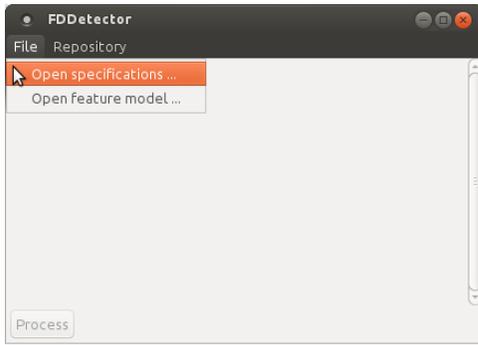


Figure 6. The Main Interface of the Automated Tool.



Figure 8. The Specification of the new evolution.

gives the possibility to load it into the repository. Initially, the creation of the repository was done manually, but its update can be automatic to load the new features from specifications or application feature models.

be able to bind these variants with a variation point from the repository.

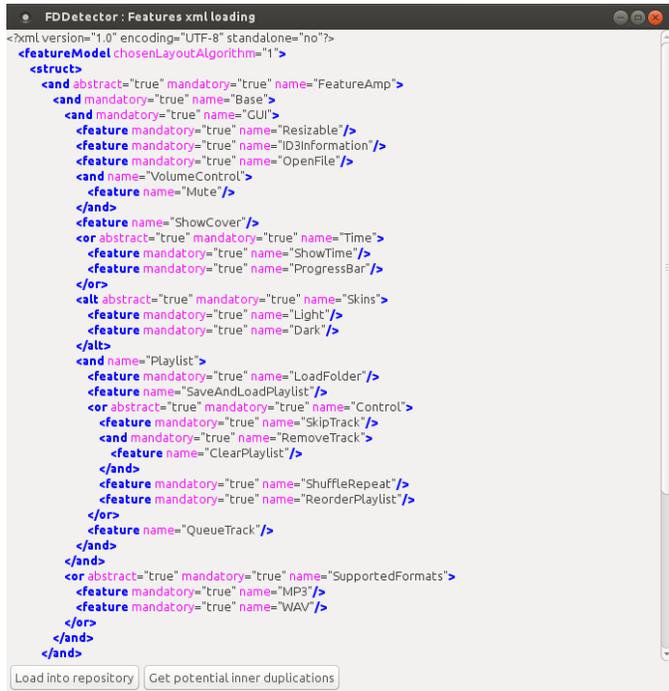


Figure 7. The Feature Model Loading.

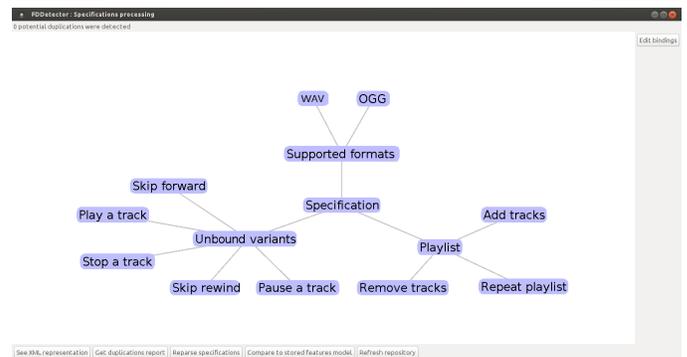


Figure 9. The graph corresponding to the specification.

This interface provides other features such as the display of the generated XML, the re-processing of the specification after a modification of the repository, the comparison between the specification and the feature model to detect duplication between them, the refreshment of the repository after the binding of new features, and the generation of the log.

B. The Specification

We consider the specification of a new evolution of the product line. The specification is displayed in Figure 8. It contains new features to be introduced in the application, but we added intentionally other duplicate features that already exist in the domain model.

When we press the button "Process", the specification is processed and transformed into an XML document. In Figure 9 is presented the graphic form of the generated XML. This presentation facilitates the visualization of the different new features introduced by the specification. In addition, the graph distinguishes the new variants added by the specification by relating them to the node "unbound variants". The user will

C. The Duplication Detection

The graph generated for the specification facilitates the distinction of duplicate features inside the specification by presenting them in a different color. For our test, the specification does not contain duplications as displayed in the top left corner of the interface.

If we want to search duplications between the specification and the feature model, we use the button "Compare to feature models". The result of this operation is illustrated in Figure 10. Two duplications are detected in the domain model, which represents 20% of the features in the specification. The percentages calculated in this interface will allow us to estimate the gain in the development cost. To visualize the details of the detected duplications, we can generate the log.

In this test, we detected the duplication between the specification and the domain model, but the same test can be performed between the specification and the application

Feature model	Number of duplications	% duplications	View details	Generate log
Domain model FeatureAMP	2	20%	View details	Generate log

Figure 10. The result of duplication detection.

model. Besides, it has to be noted that in this paper, we focused on the evaluation of the efficacy of the proposed framework, which means, whether the framework allows the detection of duplication or not. Ongoing work consists of testing the effectiveness of the solution by applying it to a complex product line and carrying out a quantitative evaluation to estimate the effort gained by using the proposed approach.

VII. RELATED WORK

In this section, we provide an overview of the studies most relevant to our work by categorizing them according to the issues addressed in this paper.

A. Evolution of Feature and Variability Models

A plethora of studies have dealt with evolution of feature and variability models. For instance, in order to reduce complexity and improve the maintenance of variability in large-scale product lines, Dhungana et al. [34] proposed a method to organize product lines as a set of interrelated model fragments that define the variability of particular parts of the system, and presented a support to semi-automatically merge the different fragments into a complete variability model. The same approach was proposed by Pleuss et al. [35] for feature models.

White et al. [36] presents a new approach for handling feature model drift, which represents the problem of introducing one or more changes in a feature model's constraints. For this, they propose a technique called MUSCLES that consists of transforming multi-step feature configuration problems into Constraint Satisfaction Problems (CSPs), then uses a constraint solver to generate a series of configurations that meet the multi-step constraints.

Cordy et al. [37] defined two particular types of features, regulative features and conservative features, and explained how the addition of these features to the SPL can reduce the overhead of model-checking.

The common denominator of the cited studies is that they all consider evolution in domain engineering, while our approach deals with evolution in application engineering.

B. Model Defects in SPL

Several papers in the literature have addressed model defects caused by SPL Evolution. For example, Guo and Wang [38] proposed to limit the consistency maintenance to the part of the feature model that is affected by the requested change instead of the whole feature model.

Romero et al. [39] introduced SPLEmma, a generic evolution framework that enables the validation of controlled SPL evolution by following a Model Driven Engineering approach. This study focused on three main challenges: SPL consistency

during evolution, the impact on the family of products and SPL heterogeneity.

In [40], Mazo provides a classification of different verification criteria of the product line model that he categorizes into four families: expressiveness criteria, consistency criteria, error-prone criteria and redundancy-free criteria. Redundancy can easily be confused with Duplication, but it is completely different, because Mazo focuses on redundancy of dependencies and not redundancy of features. The same study defines also different conformance checking criteria, among which two features should not have the same name in the same model. This is also different from our approach, which is based on equivalence and not only equality of features.

In order to locate inconsistency in the domain feature model of a SPL, Yu [41] provides a new method to construct traceability between requirements and features. It consists of creating individual application Feature Tree Models (AFTMs) and establishing traceability between each AFTM and its corresponding requirements. It finally merges all the AFTMs to extract the Domain Feature Tree Model (DFTM), which enables to figure out the traceability between domain requirements and DFTM. Using this method helps constructing automatically the domain feature model from requirements. It also helps locate affected requirements while features change or vice versa, which makes it easier to detect inconsistencies. However, this approach is different from our own one, because we suppose that the domain and application models exist, our objective is hence to construct a more formal presentation of them to facilitate the search of the new features in these models.

Kamalrudin et al. [42] use the automated tracing tool Marama that gives the possibility to users to capture their requirements and automatically generate the Essential Use Cases (EUC). This tool supports the inconsistency checking between the textual requirements, the abstract interactions and the EUCs. Unlike our approach, this one focuses on use cases instead of feature models.

Another approach proposed by Barreiros and Moreira [43] consists of including soft constraints in a feature model, which brings additional semantics that allow improved consistency and sanity checks. Hence, a framework is provided that injects soft constraints into publicly available feature models and recreates typical patterns of use. These features are then subjected to automated analysis to assess the prevalence of the proposed method. While this approach deals with constraint inconsistency, our own one focuses on feature duplication.

C. Evolution in Application Engineering

In the literature, the evolution in domain engineering have been discussed in many papers, while few studies focuses on application engineering. Among these studies, Carbon et al. [44] presented an empirical study, which consists of adapting the planning game to the product line context in order to introduce a lightweight feedback process from application to family engineering at Testo, but it does not provide a general approach that is applicable to all SPLs.

Hallsteinsen et al. [45] introduced the concept of Dynamic Software Product Lines (DSPL), which provides mechanisms

for binding variation points at runtime in order to keep up with fluctuations in user needs. This approach does not explain in details how the variability is managed between application and domain engineering.

Thao [46] proposed a versioning system to support the evolution of product lines and change propagation between core assets and derived products. However, this study also does not provide a method to manage features in application engineering.

A novel approach proposed by [47] analyses the co-evolution of domain and application feature models. It is based on cladistics classification used in biology to construct the evolutionary trees of the different models, then compares the trees using mathematical analysis and provides an algorithm to restore the perfect co-evolution of the software product line and its products.

Our approach is different from the cited approaches because it provides a feature-oriented approach to manage the evolution of derived products in a way that ensures non-duplication in the SPL feature models.

VIII. CONCLUSION AND FUTURE WORK

In the literature, many studies have addressed the evolution in SPLs, but the majority of them focused on the domain engineering phase, while application engineering has not been thoroughly discussed. Based on industrial experience, products are also likely to evolve even after their derivation, and this evolution can cause many problems especially duplication in the different artefacts of the product line. In most software engineering projects, evolutions are written in the form of natural language specifications because it is the simplest way for customers to express their requirements. At the aim of avoiding duplication when introducing the new features of these specifications into the existing SPL feature models, we proposed in this paper a framework with two main objectives. The first objective is to transform the feature models and the specifications to a more formal representation, and the second objective is to apply an algorithm that compares the new features proposed in the specifications with the features of the existing models in order to detect feature duplication. At the aim of instantiate the framework, we have started the development of an automated tool whose architecture was described in this paper. The evaluation of this tool was performed using the FeatureAMP product line.

In a future work, we intend to apply the framework on a large scale product line, which will enable us to carry out a quantitative evaluation to prove the effectiveness of our solution.

REFERENCES

- [1] A. Khtira, A. Benlarabi, and B. El Asri, "Towards Duplication-Free Feature Models when Evolving Software Product Lines," Proc. 9th International Conference on Software Engineering Advances (ICSEA'14), Oct. 2014, pp. 107-113.
- [2] K. Pohl, G. Böckle, and F. Van Der Linden, *Software Product Line Engineering Foundations, Principles, and Techniques*, Berlin, Germany: Springer-Verlag, 2005.
- [3] N. H. Madhavji, J. Fernandez-Ramil, and D. Perry, *Software Evolution and Feedback: Theory and Practice*, John Wiley & Sons, 2006, ISBN 978-0-470-87180-5.
- [4] Y. Xue, Z. Xing, and S. Jarzabek, "Understanding feature evolution in a family of product variants," Proc. 17th Working Conference on Reverse Engineering (WCRE'10), IEEE, Oct. 2010, pp. 109-118.
- [5] S. Urli, M. Blay-Fornarino, P. Collet, and S. Mosser, "Using composite feature models to support agile software product line evolution," Proc. 6th International Workshop on Models and Evolution, ACM, Oct. 2012, pp. 21-26.
- [6] A. Ahmad, P. Jamshidi, and C. Pahl, "A Framework for Acquisition and Application of Software Architecture Evolution Knowledge," ACM SIGSOFT Software Engineering Notes, vol. 38, no. 5, Sept. 2013, pp. 65-71.
- [7] C. Seidl, F. Heidenreich, and U. Assmann, "Co-evolution of models and feature mapping in Software Product Lines," Proc. SPLC'12, ACM, New York, USA, 2012, Vol. 1, pp. 76-85.
- [8] J. Holtmann, J. Meyer, and M. von Detten, "Automatic validation and correction of formalized, textual requirements," In 4th International Conference on Software Testing, Verification and Validation Workshops (ICSTW'11), IEEE, Mar. 2011, pp. 486-495.
- [9] A. Fatwanto, "Software requirements specification analysis using natural language processing technique," In 2013 International Conference on QIR (Quality in Research), IEEE, June 2013, pp. 105-110.
- [10] M. G. Ilieva and O. Ormandjieva, "Automatic transition of natural language software requirements specification into formal presentation," In *Natural Language Processing and Information Systems*, Springer Berlin Heidelberg, 2005, pp. 392-397.
- [11] P. Clements and L. Northop, *Software Product Lines - Practices and Patterns*, Boston: Addison-Wesley, 2002.
- [12] K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Technical Report CMU/SEI-90-TR-21, Carnegie Mellon University, Software Engineering Institute, Nov. 1990.
- [13] C. Salinesi, R. Mazo, O. Djebbi, D. Diaz, and A. Lora-Michiels, "Constraints: the Core of Product Line Engineering," Proc. RCIS'11, IEEE, Guadeloupe-French West Indies, France, May 19-21, 2011, pp. 1-10.
- [14] A. Hunt and D. Thomas, *The pragmatic programmer: from journeyman to master*, Addison-Wesley Professional, 2000.
- [15] B. Meyer, "On formalism in specifications," IEEE Software, vol. 2, no. 1, pp. 6-26, 1985.
- [16] G. Lami, S. Gnesi, F. Fabbri, M. Fusani, and G. Trentanni, "An automatic tool for the analysis of natural language requirements," *Informatica*, CNR Information Science and Technology Institute, Pisa, Italia, Sept. 2004.
- [17] IEEE, *IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990)*, IEEE Computer Society, 1990.
- [18] K. C. Kang et al., "FORM: A feature-oriented reuse method with domain-specific reference architectures," *Annals of Software Engineering*, vol. 5, no. 1, 1998, pp. 143-168.
- [19] J. Bosch, *Design and use of software architectures: adopting and evolving a product-line approach*, New York, USA: ACM Press/Addison-Wesley, 2000.
- [20] I. Jacobson, M. Griss, and P. Jonsson, *Software Reuse. Architecture, Process and Organization for Business Success*, Addison-Wesley, ISBN: 0-201-92476-5, 1997.
- [21] S. Creff, "Une modélisation de la variabilité multidimensionnelle pour une évolution incrémentale des lignes de produits," Doctoral dissertation, University of Rennes 1, 2003.
- [22] A. Khtira, A. Benlarabi, and B. El Asri, "Duplication Detection When Evolving Feature Models of Software Product Lines," *Information*, vol. 6, no. 4, pp. 592-612, Oct. 2015.
- [23] S. Apel and C. Kästner, "An Overview of Feature-Oriented Software Development," *Journal of Object Technology (JOT)*, vol. 8, pp. 49-84, 2009.
- [24] D. Batory, "Feature Models, Grammars, and Propositional Formulas," Proc. The International Software Product Line Conference (SPLC), Springer-Verlag, vol. 3714 of Lecture Notes in Computer Science, pp. 7-20, 2005.
- [25] C. Kastner et al., "FeatureIDE: A Tool Framework for Feature-Oriented

- Software Development,” Proc. 31st International Conference on Software Engineering, 2009, pp. 611-614.
- [26] The Apache Software Foundation, “OpenNLP,” opennlp.apache.org.
- [27] A. Khtira, A. Benlarabi, and B. El Asri, “Detecting Feature Duplication in Natural Language Specifications when Evolving Software Product Lines,” Proc. The 10th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE’15), Apr. 2015.
- [28] A. Khtira, A. Benlarabi, and B. El Asri, “A Tool Support for Automatic Detection of Duplicate Features during Software Product Lines Evolution,” In IJCSI International Journal of Computer Science Issues, vol. 12, no. 4, pp. 1-10, July 2015.
- [29] The Eclipse Foundation, “SWT: The Standard Widget Toolkit”, eclipse.org/swt/ [retrieved: July, 2015].
- [30] The prefuse visualization toolkit, prefuse.org/ [retrieved: July, 2015].
- [31] JavaLite, “ActiveJDBC”, javalite.io/activejdbc/ [retrieved: August, 2015].
- [32] The PostgreSQL Global Development Group, “About PostgreSQL”, postgresql.org/about/ [retrieved: July, 2015].
- [33] SPL2go, “FeatureAMP”, spl2go.cs.ovgu.de/projects/59 [retrieved: August, 2015].
- [34] D. Dhungana, P. Grünbacher, R. Rabiser, and T. Neumayer, “Structuring the modeling space and supporting evolution in software product line engineering,” Journal of Systems and Software, vol. 83, no. 7, 2010, pp. 1108-1122.
- [35] A. Pleuss, G. Botterweck, D. Dhungana, A. Polzer, and S. Kowalewski, “Model-driven support for product line evolution on feature level,” Journal of Systems and Software, vol. 85, no. 10, 2012, pp. 2261-2274.
- [36] J. White et al., “Evolving feature model configurations in software product lines,” Journal of Systems and Software, vol. 87, pp.119-136, 2014.
- [37] M. Cordy, A. Classen, P. Y. Schobbens, P. Heymans, and A. Legay, “Managing evolution in software product lines: A model-checking perspective,” Proc. 6th International Workshop on Variability Modeling of Software-Intensive Systems, ACM, Jan. 2012, pp. 183-191.
- [38] J. Guo and Y. Wang, “Towards consistent evolution of feature models,” In. Software Product Lines: Going Beyond, Springer Berlin Heidelberg, 2010, pp. 451-455.
- [39] D. Romero et al., “SPLEMMMA: a generic framework for controlled-evolution of software product lines,” Proc. 17th International Software Product Line Conference co-located workshops, ACM, 2013, pp. 59-66.
- [40] R. Mazo, “A generic approach for automated verification of product line models,” Ph.D. thesis, Pantheon-Sorbonne University, 2011.
- [41] D. Yu, P. Geng, and W. Wu, “Constructing traceability between features and requirements for software product line engineering,” In 19th Asia-Pacific Software Engineering Conference (APSEC’12), IEEE, vol. 2, pp. 27-34, Dec. 2012.
- [42] M. Kamalrudin, J. Grundy, and J. Hosking, “Managing consistency between textual requirements, abstract interactions and Essential Use Cases,” In Computer Software and Applications Conference (COMPSAC), 2010 IEEE 34th Annual, IEEE, July 2010, pp. 327-336.
- [43] J. Barreiros and A. Moreira, “Soft Constraints in Feature Models: An Experimental Assessment,” International Journal On Advances in Software, vol. 5, no. 3 and 4, 2012, pp. 252-262.
- [44] R. Carbon, J. Knodel, D. Muthig, and G. Meier, “Providing feedback from application to family engineering-the product line planning game at the testo ag,” Proc. 12th International Software Product Line Conference (SPLC’08), IEEE, Sept. 2008, pp. 180-189.
- [45] S. Hallsteinsen, M. Hinchey, S. Park, and K. Schmid, “Dynamic software product lines,” Computer, vol. 41, no. 4, 2008, pp. 93-95.
- [46] C. Thao, “Managing evolution of software product line,” Proc. 34th International Conference on Software Engineering (ICSE’12), IEEE, Jun. 2012, pp. 1619-1621.
- [47] A. Benlarabi, A. Khtira, and B. El Asri, “An Analysis of Domain and Application Engineering Co-evolution for Software Product Lines based on Cladistics: A Case Study,” Proc. 9th International Conference on Software Engineering Advances (ICSEA’14), Oct. 2014, pp. 495-501.

User-Centered Design of a COPD Remote Monitoring Application

Experiences from the EU-project United4Health

Berglind Smaradottir, Martin Gerdes and Rune Fensli
 Department of Information and Communication Technology
 University of Agder
 N-4604 Kristiansand, Norway
 {berglind.smaradottir, martin.gerdes, rune.fensli}@uia.no

Santiago Martinez
 Department of Psychosocial Health
 University of Agder
 N-4604 Kristiansand, Norway
 santiago.martinez@uia.no

Abstract—Recent health reforms in Norway have produced changes at all levels of the health sector, bringing to light a need for technology solutions explicitly designed for enhancing end-user collaboration. Telemedicine technology can support in this context new services that enable communication across local borders, optimizing resources and increasing cost effectiveness. This study focuses on the user-centered design, iterative development and evaluation of the user interface of a mobile application for a new telemedicine service for remote monitoring of chronic obstructive pulmonary disease patients. The tablet device application was developed based on information gathered in a workshop and group interviews where the end-users, e.g., patients and health professionals, described their preferred way of interacting with the telemedicine technology. User evaluations showed positive results on the ease of use and user satisfaction regarding the interaction with the application. Application's user interface refinements were made iteratively through several end-users' evaluations, resulting in a fully developed system suitable for remote monitoring of chronic obstructive pulmonary disease patients. Furthermore, the process led to the deployment of a telemedicine system, adopted by the partners of the project United4Health as part of the 7th Framework Programme for Research of the European Union.

Keywords—user-centered design; telemedicine; software development; usability evaluation.

I. INTRODUCTION

Health care services involve heterogeneous user groups, such as health professionals, administrative employees and patients. However, these groups share a common need: easy-to-use systems that support collaboration and coordination between users. User-centered design (UCD) has proven to be an effective methodology to identify needs across different user groups and to include them in the implementation of information and communication technology (ICT) systems [1] while increasing the usability [2] [3] and user satisfaction of clinical systems.

In Norway, a recent health reform [4] urged health organizations to implement structural changes and new pathways for citizens. Services that traditionally were offered by specialized national and regional health care institutions (e.g., follow-up of chronic diseases managed by hospitals) were transferred to primary health care managed by

municipalities. This service responsibility shift brought to light the need for an effective coordination and improved communication across borders of health care services [5][6][7], where ICT could play an essential role.

The prevalence of chronic diseases is increasing and chronic obstructive pulmonary disease (COPD) is predicted to be the fourth most fatal disease globally in 2030 [8]. COPD patients suffer from exacerbations with frequent admissions to hospital, leading to a reduced quality of life [9] and an increase of medical expenses for the society [10]. In this context, the 7th Framework Programme for Research of the European Union (EU FP7) funded the research project United4Health [11], to develop technology for remote monitoring of chronic diseases and communication across the different levels of health care services. In particular, the Norwegian contribution to the United4Health project focused on the development of telemedicine technology that supported remote monitoring of COPD patients after hospital discharge [12]. Research evidence showed that COPD patients are at an increased risk of readmission to hospital within 12 months [13][14] after hospital discharge. In the Norwegian health system, municipal health care services are responsible for patients after hospital discharge, which requires a close collaboration with general practitioners (GPs) and specialists at hospital to provide continuity of care for patients with chronic conditions. The aim of the project was then to evaluate the benefits of using ICT for monitoring COPD patients that traditionally have not had the possibility of reporting their symptoms and health status after hospitalization. The potential benefits would include reduction of hospital readmission rates with their correspondent diminution in cost and benefits on quality of life.

Two developments were made connected with the U4H project: a mobile telemedicine application for continuous monitoring COPD patient's symptoms and an information system (IS) for the new telemedicine centre through which health professionals would remotely attend the patients [1][15]. This paper presents the development of the mobile telemedicine application on a tablet device for remote monitoring of blood oxygen saturation (SpO₂) and pulse measurements. In addition, the application included a questionnaire for daily self-evaluation of COPD symptoms. Through the application, patients were able to take measurements at home that were wirelessly transmitted to

the telemedicine centre. In order to achieve acceptable levels of effectiveness, efficiency, and satisfaction, a UCD process led by a multidisciplinary research group with ICT and health background was employed for the development and evaluation of the mobile telemedicine application. The application was designed with the active involvement of end-users: patients from the patient's union of cardiac and pulmonary patients and health professionals from the municipality and partner hospital. The results from the UCD and evaluation process of the mobile telemedicine application were validated from operational and qualitative usability perspectives. The following research questions (RQ) were addressed:

RQ1: "How can a mobile telemedicine application for remote monitoring of COPD patients be developed with the contribution in the design process of patients and disease-related health professionals?"

RQ2: "What lessons from this study are transferable and applicable for the development of useful technology for other chronic disease clinical pathways?"

Following this introduction, Section II gives an overview of the research background about UCD. Section III outlines the research methodology employed and Section IV describes the results of the mobile application development. In Section V, the results are discussed and, in Section VI, the conclusion and future work are presented.

II. RESEARCH BACKGROUND

Telemedicine can be defined as a remote electronic clinical consultation using technology for the delivery of health care and the exchange of information across distance. Telemedicine covers a diverse spectrum of technologies and clinical applications [16][17][18]. Telemedicine has the potential to improve the equity of access to health care services and, therefore, the quality of the health care [17]. Mobile technology is used nowadays for multiple purposes in health, such as monitoring diseases and personalized management. Portable devices allow collection of data from patients and electronic data transmission over the Internet. Mobile networks support interactive communication between health care professionals and enable remote direct feedback to the patient. These uses are targeted at improving long-term cost-effectiveness, real time monitoring, shortening feedback's time and reducing the number of hospital visits [19].

Telemedicine systems often involve the interaction between multiple user groups through a digital system, e.g., a patient at home communicates using a device with nurse in a telemedicine or health centre, or with GP at their office. Communication in these use scenarios is usually multimodal, that is, synchronous (e.g., videoconference) and asynchronous (e.g., data transmission and dispatch); what makes it crucial to know between whom, how and when the information transmission and personal communication occur. Thus, an effective telemedicine application requires a detailed analysis of end-users' needs to inform system designers and the usability is necessary for the continuous, efficient and satisfactory use of an application. In system development, the approach of UCD [20][21][22][23]

involves end-users in all the stages and helps to understand users' needs and the context of use, which are key elements for the construction of a system framed within a clinical workflow [24]. In addition, the usability evaluation allows to analyze user's interaction and user satisfaction with the system [25][26][27].

UCD has already been used in health contexts. For instance, Martínez-Alcalá et al. [28] presented a study of telemedicine systems' development based on UCD. The aim was to develop two intuitive and efficient systems, with an optimized design of the user interface (UI) according to users' needs. The eMental System and the e-Park System development was composed of four phases: analysis, design, implementation and evaluation. They concluded that researchers and system developers must work together to integrate the knowledge of UCD towards new systems customized to users' specific needs. Further, they identified 4 research lines: (1) deployment of other telemedicine systems based on their framework including other technology; (2) development of tailored versions of a telemedicine system for mobile devices; (3) implementation of their approach in the treatment and rehabilitation therapy file; (4) incorporation of intelligent agents in telemedicine systems to support the patient and medical staff.

Ho et al. [29] described the application of a UCD process of a new remote consultation system for use in developing regions with methods such as semi-structured interviews, participant observation, and focus groups. Paper prototyping was used in the initial iterative design. De Vito Dabbs et al. [30] described the UCD of a Pocket PATH, a handheld PC that allowed lung transplant patients with data recording, messaging and decision-support to promote self-care and communication to their transplant team in hospital. The UCD process is described with the use of an interdisciplinary team in order to understand the patient users. Representative patients were recruited for meaningful selection of tasks and participation in platform for development. The evaluation was carried out in laboratory settings to measure usability, and afterwards, completed by an assessment of the functionality through a field study. Das et al. [31] used a co-design approach to involve users in the design process. Users were COPD patients that explored mobile technologies to support their health condition and disease. The examples listed above show the importance of user participation from the early stage of designing a technological solution. However, many studies like these did not reach final deployment stage. The contribution of this paper is a case study with a UCD process of a COPD remote monitoring application describing all the stages of design, whose final result has been deployed in real settings.

III. METHODOLOGY

Qualitative methods such as observations and group interviews were used for data collection and analysis during the UCD process of the telemedicine tablet application, which was framed within the research project United4Health [11][12]. The UCD process was executed in two phases with a total duration of 6 months during 2013 and 2014. The process is described in Figure 1: (A) workshop with

representative end-users, such as patients and health professionals; (B) iterative design of the tablet application for COPD remote monitoring. Each sub-phase's output informed the input of the next. The iterative system development included a sequence of four concatenated stages: design and implementation, functional test, user evaluation and field trial.

The running commentary gathered during the two phases of the UCD process resulted in 18 hours of audio-visually recorded data, verbatim transcribed by the researchers. Transcripts were coded into categories through a qualitative content analysis [27] with the software QSR NVIVO v10 [32].

A. Workshop with End-users

A one-day workshop with 7 end-user representatives (e.g., patients, health professionals and technicians) was hosted by the University of Agder, Norway. The aim was to understand the context of use and to work out the user requirements for the design of the tablet application for remote monitoring. In addition, the workshop was a source of information and familiarization for end-users with the research team and health professionals working in the

project. The participants were 2 members of the union of cardiac and pulmonary patients, mean age of 69 years; 2 nurses and 1 project-leader from the municipality and hospital, mean clinical experience of 6 years with COPD patients; and 2 technicians from hospital responsible for correct functioning and maintenance of the tablet devices, mean of 6 years of experience working with medical technical equipment.

The workshop lasted 5 hours and was divided into two parts. In the first part of the workshop, participants were given an introduction to the research project United4Health. A prototype demonstration of wirelessly transmitted measurements of SpO₂ and pulse was shown to end-users on a tablet device to facilitate the understanding of the context of use of the system. Additionally, a videoconference between a patient and a health care professional was tested. The members of the union of cardiac and pulmonary patients described their preferred way of interacting with the application at home and suggested ideas for the UI's layout. The participants used colorful post-it notes and handmade sketches to describe application's functionalities and design.

In the second part of the workshop, participants described their suggestions for the procedure of remote monitoring of a COPD patient, such as taking measurements at home, transmitting measurements' values through the system to the telemedicine centre and illustrating the feedback given from telemedicine centre to a COPD patient at home.

B. Iterative Design

The design of the application was carried out through the iterative execution of the following stages: design and implementation, functional test, user evaluation and field trial. A development team supervised by one of the researchers developed the system. An interaction designer hired by the team was in charge of the initial graphical user interface and interaction design.

1) *Design and Implementation*: The results from the workshop led the initial design and implementation of a Java native application. Java includes libraries for several low-level application program interfaces (APIs), in particular for the Bluetooth connectivity and communication with sensor devices. In addition, using Java allowed the application to be used across different tablet devices. The outcome of the subsequent sub-phases informed additional user requirements included in the implementation of the user interface design (UID) and system's functionality.

2) *Functional Test*: The facilities of the Centre for eHealth and Health Care Technology of the University of Agder, Norway, were used as a test bed for a functional test of the implemented application. It allowed verifying whether the system matched the requested functionality determined by users in the workshop and in user evaluations from other iterations. Performance and scalability of the system were not within the scope of the functional test.

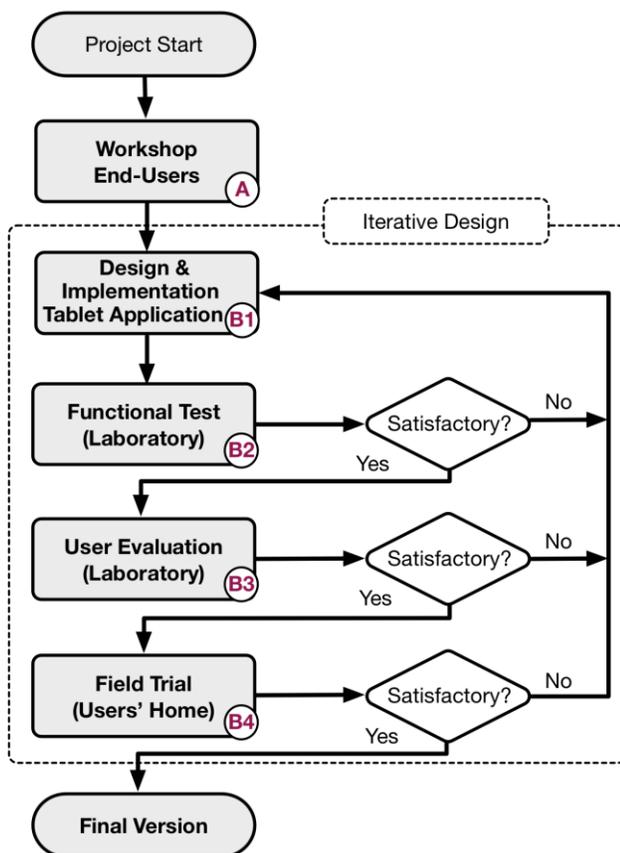


Figure 1. The User-Centered Design Process.

3) *User Evaluation*: Two evaluations of the application's prototype were carried out with end-users in the Usability Laboratory at the Centre for eHealth and Health Care Technology. The facilities had two separate test rooms (referred to as "test room 1" and "test room 2") and one observation room. The infrastructure is further described in [33]. The user evaluations had the aim to provide end-user's feedback to the development team about system's errors and potential refinements. They consisted of a series of tasks using a think aloud protocol [34][35][36]. Group interviews were made at the end of the evaluations to complete the feedback.

a) *Evaluation 1*: In total 15 end-users participated in the first evaluation. They were: 13 nurses and physicians from municipality and hospital partner and 2 technicians from hospital partner. During the test, the participants were involved in a role-play scenario. In the patient's home (represented by test room 1), health care professionals simulated the patient's use of tablet application (see Figure 2). At the same time, the telemedicine centre (represented by test room 2) contained the health care professionals that interacted with patient's home. The functionalities tested at a patient's home consisted of taking and sending patient's measurements (i.e., SpO₂ and pulse), filling and sending a questionnaire to the telemedicine centre. In addition, a videoconference session between the patient and the telemedicine centre was evaluated. There were three repetitions of the scenario with different users. The overall duration of the evaluation was 6 hours.

b) *Evaluation 2*: The second evaluation included another role-play with the new telemedicine application. It was carried out two weeks after the first evaluation and included 9 end-users: 2 members of the patient's union (who played the patient's role), 3 nurses from municipality (who played telemedicine centre health professional's role), 2 nurses from hospital and 2 technicians from hospital. The test simulated the following interactions with the application: (1) user training of COPD patient in hospital with instructions from a hospital nurse; (2) COPD patient at home taking measurements, filling in symptom self-evaluation questionnaire and sending it to the telemedicine centre; (3) videoconference between COPD patient at home and a health professional at the telemedicine centre. There were two iterations of the user evaluation, with a total duration of 5 hours.



Figure 2. End-user testing the tablet application during the evaluation.



Figure 3. The remote monitoring equipment.

4) *Field Trial*: A field trial was carried out with 6 diagnosed COPD patients (mean age 72.6 years). They tested the continuous functioning and interaction with the technology at home during a period of 7 days. The trial was performed across several weeks, lasting 5 weeks in total. Each participant was equipped with a suitcase including a pulse oximetry device (Nonin Onyx II, 2012) and a tablet device (Lenovo ThinkPad tablet 2, 2013, Windows 8.1) with the telemedicine application installed. In addition, an adjustable USB camera and a headset were included for the videoconference. Figure 3 shows the remote monitoring equipment. Every day, the participants used the tablet application for measurements with the pulse oximetry device filled in the symptoms' self-evaluation questionnaire. The data were sent over the mobile network to the telemedicine centre. A videoconference session between the participant at home and a health professional at the telemedicine centre was tested in addition.

All these tasks were performed using the tablet device. After each week of testing, the research team visited each

participant at home and made a user evaluation of the application and an interview. The user evaluation entailed switching on tablet, logging in to the telemedicine application, taking measurements, filling in symptom self-evaluation questionnaire, sending the data to the telemedicine centre and answering a videoconference call from the telemedicine centre. The interviews focused on the user experience and suggestions for further improvements. The users' suggestions in the field trial were incorporated in the iterative refinements of the tablet application. More details on the field trial are presented in [37].

IV. RESULTS

The results were obtained from the content analysis of the transcripts of the audio-visually recorded data and annotations and observations during the UCD process. To ease the reading, the results of each phase are separately presented.

A. Workshop with End-users

The contributions from end-users in the workshop are grouped in 3 different categories: context of use, user interface design and procedure for remote monitoring.

1) *Context of Use*: Patient representatives explained that their individual's level of physical energy was regularly low and even simple actions, such as using a tablet device, might become unachievable. This issue underlined the importance of designing an easy-to-use application that did not require much physical effort and mental workload to be successfully used. Therefore, it was suggested that user interaction with the system must be minimal, with only the few necessary actions. One participant stated: "Usability is extremely important for the interaction with this application since COPD patients have little energy left on bad days".

2) *User Interface Design*: Patients agreed with the authentication method through a personal identification number (PIN) mechanism, although they expressed having difficulties remembering numbers and they preferred to be able to choose their own PIN instead of using a pre-defined one. In addition, they requested to have the user's name at the top of the home screen after each successful login. Patients required seeing the results of their own measurements on the device's screen before sending them to the telemedicine centre. They asked for receiving immediate feedback when measurements were successfully delivered. A time-span visualization of several days of measurement results was also suggested where patients could see measurements from previous days. Another request was the possibility of seeing the health professional through a videoconference to simultaneously guide the patient through any of the tasks.

For the interface's layout, patients chose not to have nested menus (e.g., one patient representative said: "you cannot ask elderly people to remember what is inside each menu") and instead, only one touch area per action. Suggestions included 6 squared big-size touch areas, with readable and appropriate function's names. The 3 most important functions were placed at the top: "new measurements", "daily questionnaire" and



Figure 4. Users' suggestions for the UI of the tablet application's main screen.

"videoconference". The other 3 touch areas with less frequently used functions were placed at the bottom: "historical data", "information about COPD", and "user instructions", see Figure 4. Further, it was concluded that the system was not to be used for emergency situations, so a written text was displayed that said "Call 113 for emergency" was suggested.

For the questionnaire, end-users suggested multiple touchable selections for the daily self-evaluation of symptoms. Specifically, to have six questions visible on the screen at the same time because patients were afraid they would get tired of reading the questions one by one (see Figure 5). The button to navigate to the next step, labelled "Next", had to be visible at the bottom of the screen. The users requested to be able to review the questionnaire answers before sending the self-evaluation questionnaire.

3) *Procedure for Remote Monitoring*: One of the most important findings of the workshop was the description of the procedure for the use of the telemedicine application for remote monitoring of COPD patients.

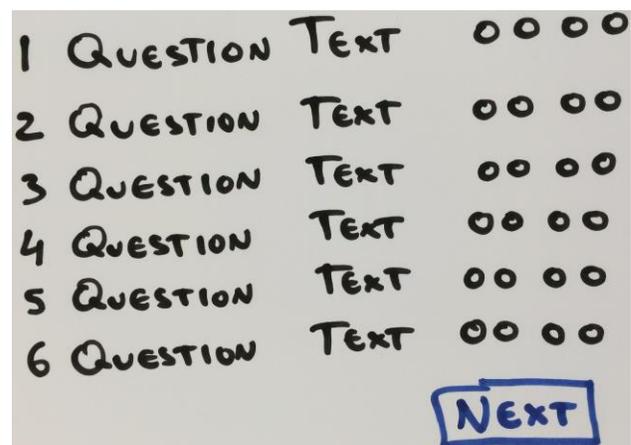


Figure 5. User's UI suggestions for the questionnaire for daily self-evaluation of symptoms.

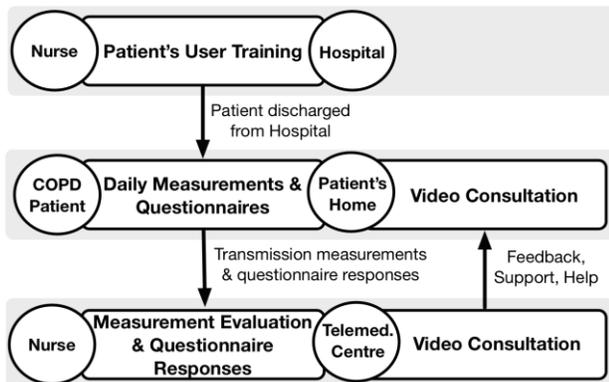


Figure 6. Procedure for remote monitoring.

Figure 6 shows the end users' suggestion for the process and feedback in the remote monitoring scenario. In addition, instructions were required to be concise and to be additionally available on paper and through the system.

It is a common practice in a given telemedicine centre to differentiate patient status by an easy-to-interpret color scheme, called triage. Triage color was represented in this case by a green color for measurement values within the predefined cut-off values; yellow color for requiring attention and red one to trigger alert. Yellow and red colors were activated when measurement values were outside the predefined cut-off values. Patient representatives initially suggested that patients at home should be able to see the triage color related to their own measurements in order to have a feeling of control of their own health. However, a "false" red measurement (e.g., cold finger may alter measurement readings) could potentially increase patient's anxiety. At the end, patient representatives agreed with the option that only health care professionals could see the triage's color.

B. Iterative Design

The contributions from the iterative design are presented following the sub-phases of design and implementation, functional test, user evaluations and field trial.

1) *Design and Implementation:* In the sub-phase design and implementation, the workshop's results were transformed into user requirements. The initial graphical user interface (GUI) for the main screen of the tablet application was outlined including the two functions "New Measurement" and "Questionnaire", which were placed at the top, see Figure 7.

For the GUI of the daily self-evaluation questionnaire, three questions with touch areas for answers were displayed with a legible text on a tablet device, see Figure 8. Outcomes from further iterations' sub-phases contributed to refine the user requirements and improve the application implementation.

Based on the initial GUI, a first prototype version was created. Figure 9 shows the first prototype version of the measurements' screen with the buttons "Measure Pulse" and "Send Pulse Value". The readings of SpO₂ and pulse are

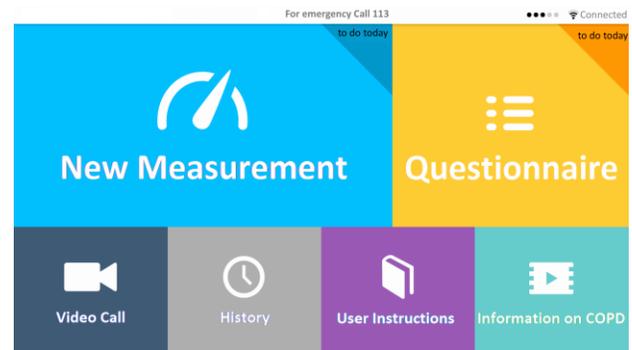


Figure 7. GUI of tablet application main screen.

shown in the right column (e.g., pulse = 85 beats per minute, and SpO₂ = 98%).

Figure 10 shows the initial prototype version of the questionnaire's UI, with one question per screen. The list of answers had to be touch-selected. A "Next" button to advance to the next question was placed under the list of answers.

2) *Functional Test:* In each iteration during the development of the application, a functionality test was run by the development team. The identification of errors at this stage proved to be relatively cost-effective to fix in terms of time and effort compared with further sub-phases.

3) *User Evaluation:* The user evaluations in laboratory settings comprised tasks to perform in the tablet application. An in-depth analysis of the observations revealed a number of usability issues. For the GUI, several problems were identified due to the insufficient text size in the UI of the measurement's screen and related to the progress bar. Some spelling errors were found in the UI wording. For the functionality, there were some technical issues related to transmission of data from the tablet device. The videoconference sound quality was insufficient, but the use of headset improved the communication. Further, while the measurement reader device showed correct measured values, wrong ones were displayed in the tablet screen and sent to the telemedicine centre. User evaluation helped to identify these issues.

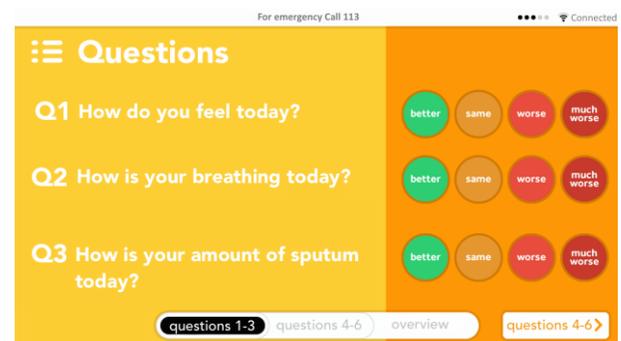


Figure 8. The GUI of the daily self-evaluation questionnaire

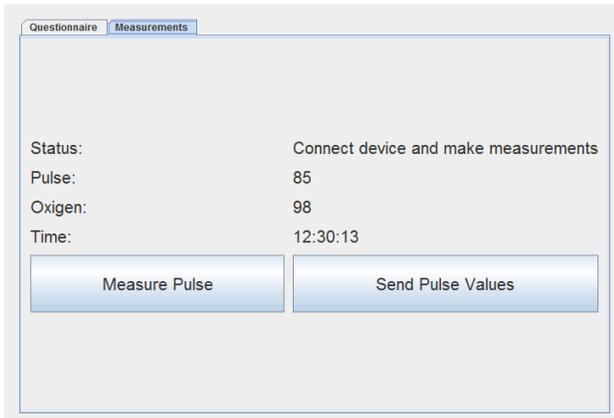


Figure 9. First prototype version of the measurement screen

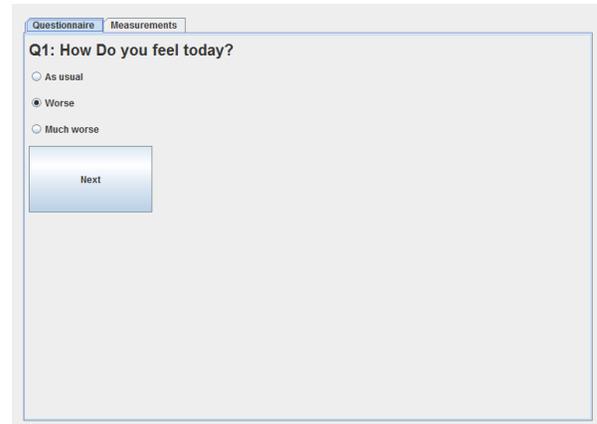


Figure 10. First prototype version of the questionnaire

In the group interviews after the evaluations, user comments about the tablet use were overall positive. They refer to the usability of the application and its functionalities: “I think this will help us if we get worse; the tablet was easy to use with 5 or 6 functions and few things that should be touched to do measurements”. Comments also addressed the feeling of safety after using the system for few days in a row: “This is a fantastic procedure and a nice service for COPD patients. Initially I was skeptical because I was afraid this would be too technical and little human, but now I think this will give patients a feeling of safety, especially the first 14 days after hospital discharge”. Other comments referred to the need of user training: “With some user training I think most people could use this, it was not complicated. If you forget how to do it, you can contact telemedicine centre”. Patients also positively commented about the videoconference: “It was a good feeling to have the videoconference with telemedicine centre. I think it is good to see and hear the nurse for users at home”. About the interaction with the tablet device, one of the patients stated: “I assume finger interaction will work well for most elderly people”.

The tablet application went through several iterative refinements to implement the findings from the user evaluations. These refinements included the display of the questionnaire with the adequate number of questions per screen, reduced from 3 in the initial GUI design to finally 1 per screen in the final implementation to ease the individual reading. A review of the questionnaire’s answers was included to allow patient to double check the filled-in answers before sending them in. Initially, a progress bar notified data transmission but it was unclear for distinguishing between successful and unsuccessful data delivery. A feedback notification pop-up window was shown, displaying a round face with an associated color code (i.e., green smiley face for successful delivery and red sad face for unsuccessful one). In addition, the user manual with intuitive images to guide step-by-step how to handle the measurement devices was requested. In this line, the GUI

corresponding to the new measurement was improved by reducing the information load to perform tasks.

4) *Field Trial*: The usability evaluations performed during the field comprised 4 tasks with associated sub-tasks and several usability problems were revealed. In the GUI of the measurements’ screen, the text “New Measurement” was used twice, as a heading but also as an action bar, creating confusion on which was one had to be selected to start the action. When choosing the action bar, a pop-up window opened over the instruction text, impeding its reading. The size of the touch area to answer the videoconference call was too small. Regarding the interface design, the text size was evaluated as sufficient and the choice of colors as appropriate. The interface of the main screen, measurement and the symptom self-evaluation questionnaire were easy to understand and had sufficient contrast between the elements. In the questionnaire, the size of boxes was sufficient and the overview of filled-in answers before sending was evaluated as a positive feature. For the application’s functionality, there was a lack of notification to the user when there was a data transmission error. For instance, a progress bar showed on the screen an ongoing transmission, but without notifying whether the transmission was successful or not. In addition, the videoconference had problems with sound and video quality. Initially, the quality was rated as satisfactory, but it presented some minor sound and video problems. Only one participant rated as satisfactory the videoconference quality during the whole test. The touch area to answer videoconference call was too small.

Regarding users’ interactions with the tablet device, the double touch action was problematic because users had to apply the correct touch speed and pressure. A stylus was required in some cases. One user had forgotten the correct action for starting the application and found a way around by touching another UI area. When adjusting the camera in the videoconference, one user accidentally switched off the application twice before succeeding.

The interviews showed that all participants successfully connected the equipment by themselves at home. The instruction manual was evaluated as clear and instructive, but



Figure 11. The final version of the UI's main screen

some mismatch between the content shown in the manual and the final text and layout shown in the system had to be resolved. The main frustration expressed by participants was the videoconference problem, which was related to mobile network coverage. For the interaction with the UI, most users stated that during one week they became familiar with the correct speed and pressure for touch actions.

Based on findings from the field trial, several refinements were made in the tablet application, such as the automatic start of the application because of problems with touch initiation of the program icon (i.e., equivalent to mouse double-click). It was found that, ideally, the tablet application should report the battery level of the measurement device to the telemedicine centre and patient. The videoconference image and sound quality was improved through software configuration changes. The sound quality was improved by the selection of optimal headphones and microphone setup for the users.

The participants' overall rating of the application was satisfactory concerning all interactions with the tablet (e.g., equipment setup, device connection, measurements, questionnaire filling, data transmission, and

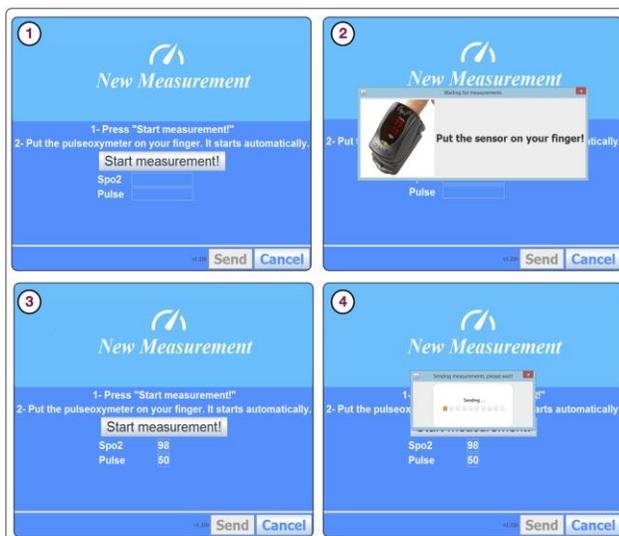


Figure 12. (1) The final version UI's of the New Measurement screen. (2) Pop-up window with instruction. (3) Readings of SpO₂ and pulse. (4) Progress bar.



Figure 13. Notification window for successful sending (left) and for unsuccessful sending (right).

videoconference). Comments referred to the design, understanding and usability of the system: "I think the application is very well designed so you do not misunderstand anything. I consider this system user-friendly"; "This application was easy to use because even an old person like me without computer experience could use it".

C. Final Version

The UCD process concluded the development of a final version of the tablet application, which was evaluated as "satisfactory" in all the sub-phases. Users started to operate the UI from the main screen of the application. The screen was divided into six differentiable touch areas with the daily functions at the top (e.g., "Questionnaire", "New Measurement" and "COPD Assessment Test". Figure 11 shows the final UI of the tablet application.

The series of steps related to the task of taking a new measurement is shown in Figure 12 and 13. The procedure included pressing the button "Start measurement" to start the operation (see Figure 12.1). When starting the measurement, a pop-up window opened and visually showed how to place the sensor on the finger (Figure 12.2). When successfully measured, the readings of SpO₂ and pulse were shown in the two fields and the button with the label "Send" would become active to send the readings to the telemedicine centre (see Figure 12.3). When pressing the "Send" button, a progress bar showed the text "Sending", representing the ongoing transmission of data (see Figure 12.4). When the data were transmitted, a feedback notification pop-up window opened to alternatively show successful or unsuccessful data delivery, see Figure 13.

The questionnaire for the daily self-evaluation of symptoms consisted of a sequence of 9 screens, 7 for the questions and 2 for reviewing and reset the answers when necessary. The question screen showed the possible answers to be touch-selected and a button with the text "Next" to continue with the remaining questions, see Figure 14 left.

The questionnaire review screen showed the answers selected and gave the possibility of resetting them when necessary. In addition, the button with the text "Send" would submit the answers to the telemedicine centre and the button labelled with "Cancel" would cancel the whole operation discarding the answers, see Figure 14 right.

Figure 14. The final version of the UI's daily self-evaluation questionnaire, question 1 (left, Q1) and answer review (right).

V. DISCUSSION

This paper has presented the UCD process for the development of a tablet device application for remote monitoring of COPD patients in home environment. Telemedicine applications typically involve multiple users in number and type, such as patients, health professionals and administrative officers. This is why the involvement of those groups of end-users in the design of a new technical application is crucial to understand the clinical workflow where the solution will be deployed, its context of use and the interactions involved. The two research questions (RQs) formulated at the beginning of this paper are answered below based on the results from the study.

About the RQ1, which asked about the development of a telemedicine application for remote monitoring of COPD patients, it has been confirmed by end-users (i.e., COPD patients and health professionals) that the employed UCD approach included their needs in the development of the application. The workshop with end-users efficiently outlined user needs, context of use and helped the user groups involved to familiarize themselves with each other and the research team. Therefore, the workshop was the key to elicit users' requirements of the application, taking on board different aspects of GUI, interaction and functionalities.

The user evaluations were carried out both in a controlled laboratory environment and at COPD patients' homes. The early evaluations in laboratory environment simulated a realistic user scenario based on constructed role-play scenario where the patients and health care professionals interacted with the technology. In addition, the laboratory provided a test environment allowing controlling the variables studied and enabled users to give feedback about GUI design and the interactions following the remote monitoring process. The laboratory test was a necessary step where to evaluate the iterations for the refinement of the application. Finally, the controlled test provided the necessary safety for, as seen in other studies, afterwards running the field trial in an optimal way [30].

The field trial allowed studying the long-term and real-time usage of the technology by COPD patients at their home and provided useful information about the interactions

between humans and technology, but also between the different technologies involved. This helped to address the common issues with interoperability [38], present nowadays in the deployment and use of telemedicine technologies [39][40].

Several lessons were learned during the study that can be transferable and applicable for technology development for other chronic clinical pathways (RQ2). In particular, intended solutions for medical environments necessarily need to firstly involve all the user groups in the creation of the solution. Secondly, the respective analysis of how this solution could best fit in an existing clinical workflow or, if non-existent, embedding the solution in a new workflow built up in collaboration with the end-user groups. Thirdly, the fact that chronic patients do not have the same levels of physical energy as healthy people underlines the importance of designing easy-to-use solutions that minimise physical effort and mental workload.

The research study of the UCD process had also some limitations such as: patient role-play by health professionals, user-scenarios tested in a simulated environment and reduced number of end-users. The health professionals took the role of the patient in the user evaluation 1 due to the low legibility of interface wording (as it can be seen in Figure 9 and 10). This was improved in the user evaluation 2, where real patients tested the interface. The simulated test environment allowed creating highly realistic scenarios under controlled conditions, and the field trial gave the opportunity to test the system in real-world settings. The number of users, despite low, meaningfully represented all the end-user groups involved [41][42].

VI. CONCLUSION AND FUTURE WORK

This study has been developed including end-users' (i.e., COPD patients and health professionals) needs, suggestions and preferences, in the design and evaluation of a COPD remote monitoring application. Positive results were reported after the evaluation in the laboratory settings, regarding ease of use of the telemedicine solution and user satisfaction. The methodology employed, UCD, transformed the end-user into a contributor of the telemedicine service design and allowing continuous refinement of the application to fully develop the system suitable for remote monitoring of COPD patients.

The telemedicine service enabled COPD patients reporting their symptoms and health status after hospitalization. The system is interoperable with other concurrent systems, resolving the common issue of interoperability present in the deployment and use of telemedicine technologies. The continuous report of symptoms for chronic patients throughout the whole health service chain together with actively including patients in building the solution, are in line with the European Union (EU) Health Strategy, "putting patients at the heart of the system and encouraging them to be involved in managing their own healthcare needs" [43]. This EU strategy aims to help current health care systems placing the patient at the centre of new treatments for chronic conditions included in the projections of global mortality for 2030 [8], such as ischemic heart disease and diabetes.

The simulation in high fidelity laboratory settings and the field trial are significant contributing factors to the ecological validity of the research here presented. In a world where human-computer interactions progressively increase in number and complexity, real-time evaluations in real-world settings become crucial to understand not only whether deployment is successful, but the efficient and continuous use of technological solutions.

Finally, the proposed UCD process has been validated by the development of a telemedicine tablet application, successfully adopted by the EU FP7 project United4Health, which focused on technologies that support remote monitoring of COPD patients after hospital discharge. As a result, 3 telemedicine centers covering 23 municipalities in Norway are currently using the final version of the application. This represents a significant contribution compared with related scientific literature where many telemedicine studies do not reach final deployment stage.

Future work will address research on appropriate identification and authentication methods for patients, more autonomous reasoning and decision support in the application, and integration of further devices to support other patient groups and clinical pathways associated with chronic diseases, such as hypertension and diabetes.

ACKNOWLEDGMENTS

The authors thank all participants and project partners for their contributions in the UCD process. They thank Joris-Jan van den Boom for the contribution in graphic and interaction design, and Baddam Sriramreddy, Mahdi Mahdavi Amjad, Navid Mokhtari, Saeed Zanderahimi, Volodymyr Shulgin and Yannick Schillinger in the development team.

This work was supported by UNiVersal solutions in Telemedicine Deployment for European HEALTH care, 2013-2015 ICT PSP (call identifier CIP-ICT PSP-2012-3) and Point-of-Care Services Agder, sub-project financed by the Research Council of Norway, 2013-15, ref 227131/O70.

REFERENCES

- [1] B. Smaradottir, M. Gerdes, R. Fensli, and S. Martinez, "User Interface Development of a COPD Remote Monitoring Application, A User-centred Design Process," The Eighth International Conference on Advances in Computer-Human Interactions (ACHI 2015) IARIA Feb 2015; pp. 57-62, ISSN: 2308-4138, ISBN: 978-1-61208-382-7
- [2] Web 3C- Web Accessibility Initiative [retrieved: November 3, 2015]. Available from: <http://www.w3.org/WAI/ redesign/ucd>
- [3] J. Chan, K. G. Shojania, A. C. Easty, and E. E. Etchells, "Does user-centred design affect the efficiency, usability and safety of CPOE order sets?," *J Am Med Inform Assoc*, 18, 276e281, 2011, doi:10.1136/amiajnl-2010-000026.
- [4] Norwegian Ministry of Health and Care Services, "The Coordination Reform, Proper treatment – at the right place and right time," Report No. 47 (2008-2009) to the Storting. [retrieved: November 30, 2015]. Available from: https://www.regjeringen.no/contentassets/d4f0e16ad32e4bbd8d8ab5c21445a5dc/en-gb/pdfs/stm200820090047000en_pdfs.pdf
- [5] Norwegian Ministry of Health and Care Services, "The Health&Care21 strategy," (HelseOmsorg21 Nasjonal forsknings- og innovasjonsstrategi for helse og omsorg) [retrieved: November 30, 2015]. Available from: https://www.regjeringen.no/contentassets/8ab2fd5c4c7746dfb51e3f64cd4d71aa/helseomsorg21_strategi_web.pdf?id=2266705
- [6] Norwegian Ministry of Health and Care Services, "The Care of the Future," (Morgendagens omsorg), Report No. 29 (20012-2013) to the Storting. [retrieved: November 30, 2015]. Available from: <https://www.regjeringen.no/contentassets/34c8183cc5cd43e2bd341e34e326dbd8/no/pdfs/stm201220130029000dddpdfs.pdf>
- [7] Directorate of Health, "National strategy for e-health (2014–2016)," (Nasjonal handlingsplan for e-helse). [retrieved: November 30, 2015]. Available from: <https://helsedirektoratet.no/Lists/Publikasjoner/Attachments/12/Nasjonal-handlingsplan-for-e-helse-2014-2016-IS-2179.pdf>
- [8] C. D Mathers and D. Loncar, "Projections of global mortality and burden of disease from 2002 to 2030," *PLoS Med*, vol. 3(11), pp. 2011-2030, Nov 2006, doi: 10.1371/journal.pmed.0030442.
- [9] T. A. Seemungal, G. C. Donaldson, E. A. Paul, J. C. Bestall, D. J. Jeffries, and J. A. Wedzicha, "Effect of exacerbation on quality of life in patients with chronic obstructive pulmonary disease," *Am J Respir Crit Care Med*, vol. 157, No.5, pp. 1418-22, 1998, doi:10.1164/ajrccm.157.5.9709032.
- [10] S. D. Ramsey and S. D. Sullivan, "The burden of illness and economic evaluation for COPD," *Eur Respir J*, 21: Suppl. 41 29s–35s, 2003, doi:10.1183/09031936.03.00078203.
- [11] United4Health 2015, European Commission Competitiveness Innovation Programme, ICT Programme to Support Policy. [retrieved: November 30, 2015]. Available from: <http://www.united4health.eu>
- [12] United4Health 2015, The Norwegian project contribution. [retrieved: November 30, 2015]. Available from: <http://www.united4health.no>
- [13] G. Gudmundsson et al., "Risk factors for rehospitalisation in COPD: role of health status, anxiety and depression," *Eur Respir J*, vol. 26, pp. 414-419, 2005, doi:10.1183/09031936.05.00078504.
- [14] I. M. Osman, D. J. Godden, J. A. Friend, J. S. Legge, and J. G. Douglas, "Quality of life and hospital re-admission in patients with chronic obstructive pulmonary disease," *Thorax*, vol. 52, pp. 67-71, Jan 1997, doi:10.1136/thx.52.1.67.
- [15] B. Smaradottir, M. Gerdes, S. Martinez, and R. Fensli, "The EU-project United4Health: user-centred design of an information system for a Norwegian telemedicine service," *J Telemed Telecare*, 5 Nov 2015, doi:10.1177/1357633X15615048.
- [16] D. A. Perednia and A. Allen, "Telemedicine technology and clinical applications," *JAMA*, 273(6), pp.483-488, 1995, doi:10.1001/jama.1995.03520300057037.
- [17] J. Craig and V. Patterson, "Introduction to the practice of telemedicine," *J Telemed Telecare*, 11(1), pp. 3-9, 2005.
- [18] R. L. Bashshur, "On the definition and evaluation of telemedicine," *Telemed J*, 1(1), p. 19-30, 1995, doi:10.1089/tmj.1.1995.1.19.
- [19] C. Bai, "Cell phone based telemedicine – brief introduction," *J of Transl Med*, 10 (Suppl 2), pp. A47, 2012, doi:10.1186/1479-5876-10-S2-A47.
- [20] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å. Cajander, "Key principles for user-centred systems design," *Behav Inf Technol*, 22(6), pp. 397-409, 2003, doi:10.1080/01449290310001624329.
- [21] K. Vredenburg, J. Mao, P. W. Smith, and T. Carey, "A survey of user-centered design practice," Conference on Human

- Factors in Computing Systems 2002 (SIGCHI 2002); p. 471-478, doi:10.1145/503376.503460.
- [22] J. Lazar, *Web Usability- a user-centered design approach*. Pearson Education, 2006, ISBN-10 0321321359, ISBN-13 9780321321350
- [23] J. Nielsen, *Usability engineering*. Elsevier, 1993. ISBN-13: 978-0125184069, ISBN-10: 0125184069
- [24] L. Goldberg et al., "Usability and accessibility in consumer health informatics current trends and future challenges," *Am J Prev Med*, Vol 40, Issue 5, Supplement 2, pp. S187-S197, May 2011, doi:10.1016/j.amepre.2011.01.009.
- [25] A. W. Kushniruk and W. L. Patel, "Cognitive and usability engineering methods for the evaluation of clinical information systems," *J Biomed Inform*, vol. 37, pp. 56-76, Feb 2004, doi:10.1016/j.jbi.2004.01.003.
- [26] M. W. Jaspers, "A comparison of usability methods for testing interactive health technologies: methodological aspects and empirical evidence," *Int J Med Inform*, vol. 78(5), pp. 340-353, 2009, doi:10.1016/j.ijmedinf.2008.10.002.
- [27] J. Lazar, J. H. Feng, and H. Hochheiser. *Research methods in human-computer interaction*. Chichester, United Kingdom, John Wiley & Sons, 2010, ISBN 978-0-470-72337-1.
- [28] C. I. Martínez-Alcalá, M. Muñoz, and J. Monguet-Fierro, "Design and customization of telemedicine systems," *Comput Math Methods Med*, vol. 2013, doi:10.1155/2013/618025.
- [29] M. Ho, R. Luk, and P. M. Aoki, "Applying user-centered design to telemedicine in Africa," *Conference on Human Factors in Computing Systems 2007 (CHI 2007) Workshop on User Centered Design for International Development*.
- [30] A. De Vito Dabbs et al., "User-centered design and interactive health technologies for patients," *Comput Inform Nurs*, vol. 27(3), pp. 175-183, May-June 2009, doi: 10.1097/NCN.0b013e31819f7c7c.
- [31] A. Das, S. Bøthun, J. Reitan, and Y. Dahl, "The Use of Generative Techniques in Co-design of mHealth Technology and Healthcare Services for COPD Patients," *Design, User Experience, and Usability: Interactive Experience Design*, pp. 587-595, Springer International Publishing, 2015.
- [32] QSR NVIVO v10. [retrieved: November 30, 2015]. Available from: http://www.qsrinternational.com/products_nvivo.aspx
- [33] M. Gerdes, B. Smaradottir, and R. Fensli, "End-to-end infrastructure for usability evaluation of eHealth applications and services," *Scandinavian Conference on Health Informatics (SHI 2014) Aug 2014*; pp. 53-59, ISSN(print): 1650-3686, ISSN(online): 1650-3740, ISBN: 978-91-7519-241-3
- [34] J. Horsky et al., "Complementary methods of system usability evaluation: surveys and observations during software design and development cycles," *J Biomed Inform*, 43(5), pp. 782-790, 2010, doi:10.1016/j.jbi.2010.05.010.
- [35] K. A. Ericsson and H. A. Simon. "Verbal reports as data," *Psychol Rev* 87, pp. 215-251, 1980.
- [36] J. Nielsen, T. Clemmensen, and C. Yssing, "Getting access to what goes on in people's heads?: reflections on the think-aloud technique," *Nordic Conference on Human-Computer Interaction (NordicCHI, 2002) Oct 2002*; pp. 101-110, doi:10.1145/572020.572033.
- [37] B. Smaradottir, M. Gerdes, S. Martinez, and R. Fensli, "Usability Evaluation of a COPD Remote Monitoring Application," *Stud Health Technol Inform*, vol. 210, p. 845-849, 2015, doi:10.3233/978-1-61499-512-8-845.
- [38] A. Shaikh, "The impact of SOA on a system design for a telemedicine healthcare system," *Netw Model Anal Health Inform and Bioinforma*, 4(1), p.1-16, 2015, doi:10.1007/s13721-015-0087-0.
- [39] R. L. Bashshur, "Guest editorial: Compelling issues in telemedicine," *Telemed E Health*, 19(5), p. 330-332, 2013, doi:10.1089/tmj.2013.9998.
- [40] R. L. Craft, "Toward technical interoperability in telemedicine," *Telemed J E Health*, 11(3), 384-404, 2005, doi: 10.1089/tmj.2005.11.384.
- [41] J. Nielsen, "Estimating the number of subjects needed for a thinking aloud test," *Int J Hum-Comput St*, 41(3), 385-397, 1994, doi:10.1006/ijhc.1994.1065.
- [42] J. Nielsen and T. K. Landauer, "A mathematical model of the finding of usability problems," *Conference on Human factors in computing systems (ACM INTERCHI'93) May 1993*, pp. 206-213, doi:10.1145/169059.169166.
- [43] European Union. *Official Journal of the European Union*, vol. C 146, 22 June 2006, ISSN 1725-2423. [retrieved: November 30, 2015]. Available from: [http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1446552106801&uri=CELEX:52006XG0622\(01\)](http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1446552106801&uri=CELEX:52006XG0622(01))

Cognostics and Knowledge Used With Dynamical Processing

Claus-Peter Rückemann

Westfälische Wilhelms-Universität Münster (WWU),

Leibniz Universität Hannover,

North-German Supercomputing Alliance (HLRN), Germany

Email: ruckema@uni-muenster.de

Abstract—This paper presents the extended research conducted for an implementation creating knowledge-supported dynamical visualisation and computation, which can also be used in combination with dynamical processing. The focus of methodologies, knowledge, and approach is data-centric, especially concentrating on cognostics and knowledge. The methodologies, knowledge resources, data structures, and workflows are very suitable for long-term multi-disciplinary context and creating application components for the use with High End Computing. The core knowledge is based on long-term knowledge resources further developed for several decades and used with many applications scenarios utilising multi-disciplinary and multi-lingual content and context like references, associations, and knowledge container collections. A major goal of the application case studies shown here is creating context-sensitive dynamical program components and algorithms from selected knowledge content. The selections are results of dynamical workflows, which are part of component implementations, e.g., including search processes and result matrix generators. Previous research has shown that long-term knowledge resources are the most important and most valuable component of long-term approaches and solutions. Here, the structures and classifications are used with independent database frameworks and programming interfaces. The results show that the methodological foundations and knowledge resources are very well suitable as long-term core base, as well as for creating dynamical application components, e.g., for visualisation and computation in multi-disciplinary, geoscientific, and spatial context. The knowledge resources can refer to any kind of resources. The overall environment allows to develop and govern extensive content structures and promote their long-term vitality. The long-term knowledge resources and cognostics are an excellent base for supporting dynamical processing sustainable, economic, long-term development of resources and components.

Keywords—*Advanced Knowledge Discovery; Universal Decimal Classification; Conceptual Knowledge; Dynamical Visualisation and Computation; Cognostics.*

I. INTRODUCTION

Many methods have been created for deploying computing and processing with knowledge discovery and dynamical visualisation. However, conceptual knowledge and cognostics supporting the creation of fully functional application components has not been considered explicitly. The target in this implementation is multi-disciplinary information with examples from geoscientific and natural sciences referring to universal knowledge and reaching into many disciplines. The secondary data utilised here can also include spatial information.

Within the last decades the value of the content, the “value of data”, has steadily increased and with this the demand for flexible and efficient discovery processes for creating results from requests on data sources. This paper presents the extended research based on the creation of knowledge-supported dynamical visualisation and computation, the results of which were published and presented at the GEOProcessing conference in February 2015 in Lisbon [1]. Applying the conceptual knowledge and cognostics in this context, this paper is especially focussing on support for data-centric approaches.

This research shows details of the latest case studies and discusses the up-to-date experiences from the implementation of the dynamical components and their integration with knowledge resources’ structures and workflows. The case studies especially consider the levels inside application components, which can be created based on universal knowledge resources and cognostics. The studies also discuss the conceptual support for generators and some resulting features.

This paper is organised as follows. Section II introduces with the state-of-the-art and motivation for this research. Sections III and IV summarise the foundations and challenges with cognostic components, Section V introduces the methodological bases, Section VI presents the previous research being a fundament for this work. Sections VII and VIII present the fundamental implementations, esp. integration referring to knowledge and dynamics, discussing the foundations, architecture, framework, integration, and dynamical visualisation and computation. Section IX discusses selected parts of the implementation and resulting components and Section X presents examples of dynamical cognostic processing in context with knowledge and cognostics. These sections show the details of the implementation and the resulting components, from geo sets, computation to index selection and some views from the resulting visualisations. Sections XI to XIII evaluate the main results, potential and summarise the lessons learned, conclusions and future work.

II. STATE-OF-THE-ART AND MOTIVATION

The creation of long-term knowledge resources and utilisation methods is one of the most pressing goals in information science as the masses of data and the loss of knowledge in society are steadily increasing in all areas. Existing projects employ segment-like spectra of disciplines in their focus. Examples are large digital libraries and projects like the Europeana [2] and the World Digital Library (WDL) [3]. For most data, source and resulting data, it is reasonable to assume

a data value at least higher than the funding value [4], which motivates for increased efforts.

As existing projects, e.g., which are only concentrating on bibliographic means, do not focus on such integration, use different and mostly isolated classifications and schemes for different areas and specific purposes. For example, there is a small number of general classifications, which are mostly used in library context. Although such classifications are used in many thousands of institutions worldwide this is neither a general use case or application scenario nor a significant share of the overall knowledge. They are missing to provide facilities for arbitrary kinds of objects, e.g., factual data and trans-disciplinary context in information science and natural sciences. Therefore, the handling of knowledge issues at a cognostic level is not reflected by processes for creation of software components. Besides these major gaps in research, shortcomings result regarding data quality, data-centric solutions, required long-term aspects, and functionality of application components.

In contrast to that, the state of the art for documentation of universal, conceptual knowledge is the Universal Decimal Classification (UDC) [5], which is one of the very few classifications providing a universal classification [6], [7]. Besides public interfaces, the implementations of the known application scenarios are not publicly available in common. Anyhow, all known scenarios have in common that they deploy only a small subset of available classifications and in the vast majority the classification process is not automated. It is necessary to develop logical structures in order to govern the existing big data today and in future, especially in volume, variability, and velocity. This is necessary in order to keep the information addressable and maintain the quality of data on long-term.

Beyond the focus of the mentioned segment-like projects the knowledge resources and concepts discussed and implemented in this research focus on the trans-disciplinary integration of arbitrary different segments and disciplines and a universal usability based on factual data and criteria. The documentation and context also integrates and refers to content and context, e.g., conceptual, procedural, and metacognitive knowledge and allows for a huge range of possible scenarios. This is a driving force to extend the use of classification in trans- and multi-disciplinary context and transfer the experiences from deploying a classification for long-term documentation and application.

III. FOUNDATIONS OF COGNOSTIC COMPONENTS

The term cognostics is related to the Latin verb *cognoscere* (con 'with', *gnoscerere* 'know, recognise'). This is in itself a cognate of the Greek verb *γινώσκω* (*gignósko*), meaning 'I know, I perceive' (noun: *gnósis*, 'knowledge') referring to 'recognize', 'conceptualize'. The archaic paradigm of many aspects of fields of human activities is, e.g., a complementary trias of animistics, empirics, and cognostics. Most cognostic content followed prehistorical and early historical times, in the later stages of development. Cognostic fundamentals have been discussed in detail by classical ancient Greek philosophers.

"Cognition" as a modern term at least dates back to the 15th century, used for "awareness" and "thinking".

Cognition is the set of all mental abilities and processes. Cognition is therefore related to knowledge, decision making [8], [9], problem solving, human-computer interaction and many more. Cognitive processes use existing knowledge and generate new knowledge.

In result, the implementation of cognostic views can be complex, not only because of application components but especially because of the knowledge-centric base, which has to be created. A lot of following researchers have picked-up the term, e.g., in context with diagnostics guided by computers [10], [11], in an early phase of data exploratory analysis and without developing a base for the knowledge itself.

The term cognostics has been more widely used with developments in information science since the nineteen-nineties. In Geocognostics refers not only to dynamical components but to different views and how to achieve this, e.g., how to integrate political and social-cultural differences, which are examples of major aspects to be considered.

The cognitive aspects of human-computer interaction are multifold, especially in the context of geographic information systems [12]. As one contribution to complex information systems, on the one hand, the accuracy of spatial databases is an important factor [13] but on the other hand it is important that researchers and users can create cognitive collages and spatial mental models [14]. Besides that, the integration of psychological aspects of spatial information [15] as well as the consideration of cultural aspects [16] has become significant for several decades. This has led to the concepts of cognitive geographic information systems [17] and the idea of geocognostics [18]. These fundamental concepts have led to implementations considering cognostics with dynamical features [19] and collaboration frameworks [20], resulting in modular cognostic component concepts [21] and knowledge based approaches and implementations [22], e.g., the new concept of object carousels.

IV. CHALLENGES WITH KNOWLEDGE AND COGNOSTICS

The insufficient care for knowledge has shown to have impact on scientific work [23] and most achievements have to be created over and over again [24] as well as on industrial developments and automation [25], [26], [27], [28], [29], [30].

Data collections can provide any huge amount of information, directly and in consequence of applied workflows. The more unstructured the data collections are the more interpretation of the data are possible. Automated processes can then easily lead to a much higher rate of misinterpretation.

Examples of misinterpretation [31] and unclear data [32], which can regularly result when not integrating high quality knowledge resources have led to suggestions and recommendations that raw data might take enormous efforts to clean, reformat, and consolidate but is very well worth. In spite of the quantity of data [33] bigger does not mean better. For many cases big data has been claimed a 'big misnomer' [34].

The quality of data cannot be neglected especially when creating sustainable structures, long-term documentations and

solutions. Some analysis considers data size being important on the one hand, e.g., volume of data. On the other hand, small volumes of data can have a large significance, examples are the following:

- Non-standard mean data like outliers can be significant for working with problems and data.
- Rare data events or attributes can be the interesting ones.
- Rare discrete values or classes.
- Missing values can lead to sparsely observed observation space.

Therefore, the target is a) to minimise the continuous rework and provide long-term features in order to document all parts of the creational steps and results and b) to integrate knowledge and cognostic features, which is a core purpose of the knowledge resources. Besides benefits, complex “all-in” folders, like in election context do show up with challenges [35], which requires alternatives.

V. METHODOLOGICAL BASES: LOGICAL STRUCTURES

To work on that goal requires to define information units and to care for depositing an appropriate segmentation in sub-units. The information units require links to the related units, e.g., superunits. The challenges are to define these structures and units for data used in different disciplinary context, in one discipline, as well as in multi-disciplinary context. These logical structures are the basic precondition for the development of functioning algorithms, which can access the units and whose application can be perfected in a next self-learning step. The tries of using unstructured data result in the fact that data volumes, variabilities, and volumes devalue the resulting values of requests. Any isolated technological approaches to the big data challenge have shown not to be constructive. A sustainable approach has to consider the data and structure itself.

- The first step is the preconditional definition of a logical, commonly valid structure for the data.
- The second step is the planning for the applications based on the logical structures in step one.
- The third step is the creation of algorithms regarding the data and data retrieval, interfaces, and workflows based on steps one and two. The fourth step is the planning of the implementation. This includes data format, platforms, and applications.

Further, the creation, development, and operation of the content and components require to consider and define the essential plans, especially:

- Plans for extending structures.
- Preparations for all required interfaces for the newly extended structures should be done.
- Plans for self-learning components.
- Plans for container formats and utilisation.
- Plans for sustainable handling of data lifecycles, data formats, and system resources.

The early stage of planning requires a concept catalogue. So far, the activities are commonly in a pre-planning stage. The

next step includes case studies on structures, algorithms, and collaborative issues (efficiency, economical cycle).

VI. FUNDAMENT FOR THIS WORK

Long-term knowledge resources can be created and used for universal documentation and re-use of content. The re-use includes discovery, as well as gathering new results and creating new applications. The knowledge resources [36] can refer to any kind of resources, e.g., to natural sciences resources or historical geographic resources [37]. Basics of knowledge organisation [38] and multi-lingual lexical linked data [39] have been discussed for various disciplines and shown the huge potential and value of the knowledge. This also shows the benefits of linking with universal classifications, especially with consequently numerical notations, which can be easily and most flexibly and efficiently used with modern applications components. Further, on the one hand, information services benefit from a comprehensive and holistic model for evaluation [40] and on the other hand, they align with the benefits for a quality management of information services [41].

The paper presents a new implementation for creating knowledge-supported dynamical visualisation and computation, which have not been integrated before for that purpose. Therefore, a major goal of the application case studies is creating dynamical program components and algorithms based on knowledge resources. The different previous projects and case studies have already shown that the combination of knowledge resources with integrated conceptual knowledge references can be used for the creation of dynamical applications.

The dynamical visualisation and computation based on knowledge resources does have numberless applications. Some prominent examples with the research presented here are knowledge discovery, visualising result matrices from workflows or search processes, and creating objects and extending knowledge resources. The framework presented here is a high level framework interconnecting several frameworks for complex system architecture, multi-column operation, and long-term creation for main resources. Therefore, the required approach is considered to be necessarily most complex from knowledge and implementation point of view.

Following the Geo Exploration and Information case studies [42] based on the actmap framework [43] a number of developments for the deployment of High End Computing resources and technologies with integrated systems are still state of the art. In addition, including the structural and conceptual knowledge based on the knowledge resources, research has been done for a different special database framework, which is as well autonomous and can be used for the creation of standalone dynamical and portable application components. The components can be integrated with the existing frameworks, as well as they can be used as standalone interactive applications.

VII. IMPLEMENTATION: INTEGRATION AND KNOWLEDGE

A. Content and context

The implementation shows the context-sensitive dynamical components based on the knowledge resources. The knowledge resources provide the structure and integrate the factual

knowledge, the references, including the references for the object classification views required for the dynamic utilisation, ensuring integration [44] and sustainability [45].

Previous case studies have shown that any suitable cartographic visualisation can be used for the presentation of the results, for example, with the Generic Mapping Tools [46] (e.g., filtering, trend fitting, gridding, views, and projections) or creating exports and imports with various products.

Most available cartographic visualisation products are too specialised in order to handle advanced knowledge workflows on the one hand and dynamical results on cognitive context on the other hand. In the presented case where the application should concentrate on the intention of presenting a special result in an abstract way we require special and flexible facilities for dynamical sketch drawings. The more, in the special case the cognitive background forbids to concentrate on detailed cartographic visualisation or mixing with modern ways of geographic conventions. Historical names, locations, and context are not adequately represented by existing modern frameworks.

Regarding both requirements for this study are fully complied by the flexibility of the implementation. The knowledge resources themselves are not restrictive regarding the use of other components for other purposes.

B. Implementation foundations

The implementation for dynamical visualisation and computation is based on the framework for the architecture for documentation and development of advanced scientific computing and multi-disciplinary knowledge [47]. The architecture implemented for an economical long-term strategy is based on different development blocks. Figure 1 shows the three main columns: Application resources, knowledge resources, and originary resources.

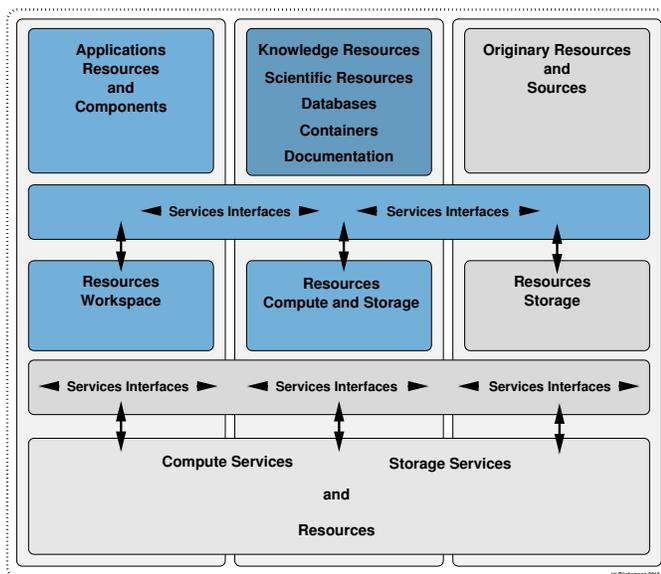


Figure 1. Architecture: Columns of practical dimensions. The knowledge resources are the central component within the long-term architecture.

The central block in the “Collaboration house” framework architecture [48], are the knowledge resources, scientific resources, databases, containers, and documentation (e.g., LX [36], databases, containers, list resources). These can be based on and refer to the originary resources and sources (photos, scientific data, literature).

The knowledge resources are used as a universal component for compute and storage workflows. Application resources and components (Active Source, Active Map, local applications) are implementations for analysing, utilising, and processing data and making the information and knowledge accessible. The related information, all data, and algorithm objects presented are copyright the author of this paper, LX Foundation Scientific Resources [36], all rights reserved. The LX structure and the classification references based on UDC [5], especially mentioning the well structured editions [49] and the multi-lingual features [6], are essential means for the processing workflows and evaluation of the knowledge objects and containers. Both provide strong multi-disciplinary and multi-lingual support.

The three blocks are supported by services’ interfaces. The interfaces interact with the physical resources, in the local workspace, in the compute and storage resources the knowledge resources are situated, and in the storage resources for the originary resources.

All of these do allow for advanced scientific computing and data processing, as well as the access of compute and storage resources via services interfaces. The resources’ needs depend on the application scenarios to be implemented for user groups.

C. Cognostics and related data

The content-supported cognostics support a plethora of knowledge object sources, information, and data features. An excerpt of examples of these features used in the case studies implemented here are:

- name,
- keywords,
- conceptual knowledge and classification,
- text,
- link references,
- dates,
- languages,
- translations,
- transliterations,
- map locations and GPS,
- various types of comparisons and context references,
- researchers’ views,
- comments,
- object specific material, classification, documentation, parameters, files, workflow descriptions, programs, . . .

All the data provided for an application, e.g., from knowledge objects in collections or containers, including textual and non-textual data, scientific data, mathematical data, technical data

as well as comments can be analysed by the workflows and application components.

The visualisation components can use all the cognitive attributes in order to further exploit the information for the respective application targets. In addition, with the multi-disciplinary long-term features and structures the knowledge resources provide the means for a sustainable integrated research data management [50].

This is significant in combination with integrated realia and digital resources, for example, in digital museums projects [51]. An example for creating components usable with complex environments is an agent-based modeling and simulation in archaeology [52]. This is especially interesting for High End Computing environments, in the given examples, for example, for archaeological simulations [53].

In the presented scenarios implemented with this research, the new resources overcome the insufficient documentation, the structural and long-term deficits. Therefore, the integration benefits from the combination of structured and unstructured data.

A lot of unstructured data is provided on the Internet. For some scenarios this source might provide the largest percentage of unstructured data, regularly integrated in the workflows, e.g., with knowledge discovery processes. Regarding non knowledge resources based sources the proprietisation (or 'Proprietarisierung' and 'proprietaryisation' in German and French a bit more concisely reflecting the stricter and essential Latin 'proprius', meaning 'own, individual, special, particular, characteristic' and dissociating from 'proper' and 'property') is not only a threat for the free Web [54], [55], [56] but also for the workflows, which can be based on unstructured data and free and open access data, at least to a flexible extend. The situation may lead to challenges with application scenarios, which contain a large percentage of references to unstructured data on the free net as well as with workflows, which require or benefit from such data.

With unstructured Big Data, which is most of the overall amount of data available, there is also a lot of bias [57]. With high end solutions and resources, e.g., with High Performance Computing, there is an increasing demand for techniques supporting fault tolerance [58]. For High End Computing solutions this can be achieved by modular structures and documentation.

One of the important features of algorithms and structures supporting these applications is the ability to cope with failures in information and applied knowledge [59]. With the facilities of the knowledge resources for precision and fuzziness the components can cover a flexible range of cognitive coverage.

VIII. IMPLEMENTATION: INTEGRATION AND DYNAMICS

A. Integration and computation

The context of the application components is fully integrated with the knowledge resources and dynamical components [60]. The screenshot (Figure 2) illustrates some features. Shown examples illustrate features of Active Source, computed and filtered views, LX information, and aerial site photographs, e.g., from Google Maps.

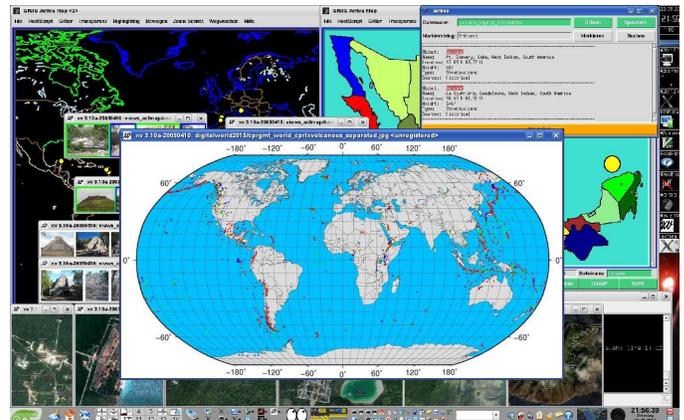


Figure 2. Dynamical use of information systems and scientific computing with multi-disciplinary and universal knowledge resources [48].

Many general aspects of dynamical use of information systems and scientific computing have been analysed with the collaboration house case studies.

B. Implemented content and application dynamics

The main groups of challenges are resulting from the content and from the applications.

- Content side: From content side, the knowledge resources provide the central repository and infrastructure ('Knowledge as a Service') for discovery and component creation.
- Application side: From application side, the dynamical components can deploy the resources to any extent and in any step of workflows.

Any part of the components and features can be assembled from the knowledge resources' workflows. As an example, the site context and factual data components, the database and the graphical user interface components including event definition and management can be dynamically created via transform routines and concatenate operations.

C. Dynamical visualisation and computation implementation

A number of different visualisation tools and frameworks have been analysed in the latest case studies. The results presented here were mostly realised with Tcl/Tk [61] for the dynamical visualisations, Fortran and C based programs for required algorithms, Message Passing Interface (MPI) [62], and Perl for dynamical scripting.

Many components have been developed for the actmap framework [43] and successfully used and verified in context with existing scenarios. Besides the actmap framework additional possibilities of creating application programming interfaces and graphical user interfaces for dynamical visualisation of knowledge matrices have to be analysed.

As a simple example for a dynamical, portable, and standalone component an application like Tclworld has been considered [63]. The application is built on a very portable Tool

Command Language (TCL) base integrating programming, database, and user interfaces.

The database application programming interface [64] is very simple, portable, and extendable. The database graphical user interface [65] can be used within the same application and is based on a rapid prototyping concept. Both interface models allow dynamical control and extension of any features regarding the application, as well as for the content.

IX. IMPLEMENTATION AND RESULTING COMPONENTS

The knowledge resources are very flexibly supporting documentation and handling of knowledge content and context. The features include cognostics as well as dynamical application support.

Key to the flexibility is that any knowledge object can carry as much documentation as required. It is the task of the application workflow to make use of the available information to the extend needed and to develop or create complementary information. Creating knowledge-supported dynamical components starts at the application to knowledge resources' interface level. The conventional components and processes have been described and discussed in practice in detail in previous research, regarding Active Map Software [43], case studies [42], and knowledge integration for classification and computation [44].

For the visualisation and computation using the matrix objects with spatial and georeferenced context, a new application instance "lxworlddynamic" has been created based on the knowledge resources and interfaces. This component re-using the Tclworld interfaces is the required extension complementary to the actmap framework components. All parts of the component shall

- support dynamical and cognostic features,
- integrate with the knowledge resources and existing components, e.g., georeferenced objects,
- be usable interactively,
- have access to the content, e.g., index selection,
- facilitate a standalone application assembly, and
- allow a flexible configuration of all aspects of the applications' visualisation and computation, including cognostic support.

Especially, the component requires a site handler database, has to refer to sets of georeferenced objects, a level handler for selecting levels of detail, to create groups of matrix objects, to support for an individualised configuration, to facilitate index selections on the generated matrix elements, to visualise the matrix elements and context graphically, and has to provide associated data textually and numerically.

Therefore, the implementations done for this study concentrate on creating dynamical, generated, cognostic components. Explicitly, the focus was not on georeferencing standards or cartographic issues or precision. Any of such features can be developed by third parties being interested in supporting their individual scenarios.

A. Knowledge resources and geo sets

The integrated information systems can generate result matrices based on the available components and workflows. The result matrix generators can be configured to deliver any kind of result matrix. One base for the implementation is the generation of georeferences data resulting from requests. The listing (Figure 3) shows an example of a result matrix, an excerpt of the generated site handler database.

```

1 {LX Site 20.687652 -88.567674} site 20.687652 88.567674
2 {LX Site 20.682658 -88.570147} site 20.682658 88.570147
3 {LX Site 20.682859 -88.568548} site 20.682859 88.568548
4 {LX Site 21.210859 -86.80352} site 21.210859 86.80352
5 {LX Site 21.097633 -86.796799} site 21.097633 86.796799
6 {LX Site 21.157199 -86.834736} site 21.157199 86.834736
7 {LX Site 21.157199 -86.834736} site 21.157199 86.834736
8 {LX Site 21.094751 -86.812248} site 21.094751 86.812248
9 {LX Site 41.377968 2.17804} site 41.377968 -2.17804
10 {LX Site 41.375842 2.177696} site 41.375842 -2.177696
11 {LX Site 38.676439 -0.198618} site 38.676439 0.198618
12 {LX Site 38.677683 -0.198103} site 38.677683 0.198103
13 {LX Site 21.234502 -86.740494} site 21.234502 86.740494
14 {LX Site 21.184412 -86.807528} site 21.184412 86.807528
15 {LX Site 16.043421 -61.663857} site 16.043421 61.663857
16 {LX Site 16.043153 -61.663374} site 16.043153 61.663374
17 {LX Site 17.633225 -63.236961} site 17.633225 63.236961
18 {LX Site 17.633225 -63.236961} site 17.633225 63.236961
19 {LX Site 51.151786 10.415039} site 51.151786 -10.415039
20 {LX Site 20.214301 -87.429103} site 20.214301 87.429103
21 {LX Site 20.493276 -87.735701} site 20.493276 87.735701
22 {LX Site 20.494663 -87.720294} site 20.494663 87.720294
23 {LX Site 20.494761 -87.720138} site 20.494761 87.720138
24 {LX Site 40.821961 14.428868} site 40.821961 -14.428868
25 {LX Site 20.365228 -87.452545} site 20.365228 87.452545
26 {LX Site 20.365228 -87.452545} site 20.365228 87.452545

```

Figure 3. Excerpt of generated site handler database (lxworlddynamic).

The database is the result of a request summarising results on objects referring to a defined context, in this case references between archaeological and geological objects.

The framework provides a number of features like level handlers and sets of object georeferences. The listing (Figure 4) shows an excerpt of the generated site level handler.

```

1 foreach i {
2   { ... }
3   { ... }
4   ...
5 } {+ $i level 2}

```

Figure 4. Excerpt of generated site level handler (lxworlddynamic).

The level handler manages the site handler database, which can also be generated and updated dynamically. Appropriate entries are managed by the geo::set. The listing (Figure 5) shows an excerpt of the generated geo set.

```

1 geo::Set {
2   { ... } site ... ..
3   { ... } site ... ..
4   ...
5 }

```

Figure 5. Excerpt of generated geo::set (lxworlddynamic).

Groups of objects, e.g., associated archaeological, geological, meteorite, and volcanological sites, as well as subgroups like pottery and stones, can be dynamically associated and handled in the generated component. The listing (Figure 6) shows an excerpt of the generated database matrix.

```

1 + {Archaeological site} : {A selected site with findings
2   of human activity, complementary to a
3   {Geological site} .
4   These sites have been dynamically created
5   from a request to the LX knowledge resources ... .
6   These sites have been ... .
7   }
8
9 + {Geological site} : {A selected site with geological
10  findings, e.g., a {Volcanological site} or a
11  {Meteorite site} , complementary to an
12  {Archaeological site} .
13  This site has been ...}
14
15 + {Meteorite site} : {A selected site with meteorite
16  findings, e.g., meteorite crater, a special
17  {Geological site} .
18  This site has been ...}
19
20 + {Volcanological site} : {A selected site with
21  volcanological findings, e.g., volcanological
22  findings like a volcano or fumarole, a special
23  {Geological site} .
24  This site has been ...}
25
26 + Pottery      : {Archaeological site} major.countries {
27   Italy France Spain Greece}
28
29 + Stone        : {Geological site} major.countries {Italy
30   France Spain Greece}

```

Figure 6. Excerpt of generated database matrix (lxworlddynamic).

Here, the matrix includes site and object types for the respective matrix with excerpts of descriptions and linked references.

B. Index selection and configuration

When a representation of matrix objects in dynamical spatial cartographic context is possible then selected objects can be integrated either from the matrix elements or from the context elements and references (Figure 7).

```

1 foreach i [geo::Names] {
2   if {[lindex $geo::db($i) 0]=="city"} {
3     "LX_World_database" $i : {city} loc [lrange $
4     geo::db($i) 1 end] }
5   if {[lindex $geo::db($i) 0]=="mount"} {
6     "LX_World_database" $i : {mount} loc [lrange $
7     geo::db($i) 1 end] }
8   if {[lindex $geo::db($i) 0]=="site"} {
9     "LX_World_database" $i : {site} loc [lrange $
10    geo::db($i) 1 end] }
11  if {[lindex $geo::db($i) 0]=="lake"} {
12    "LX_World_database" $i : {lake} loc [lrange $
13    geo::db($i) 1 end] }
14  if {[lindex $geo::db($i) 0]=="road"} {
15    "LX_World_database" $i : {road} loc [lrange $
16    geo::db($i) 1 end] }
17  if {[lindex $geo::db($i) 0]=="rail"} {
18    "LX_World_database" $i : {rail} loc [lrange $
19    geo::db($i) 1 end] }
20  if {[lindex $geo::db($i) 0]=="river"} {
21    "LX_World_database" $i : {river} loc [lrange $
22    geo::db($i) 1 end] }
23  if {[lindex $geo::db($i) 0]=="grid"} {
24    "LX_World_database" $i : {grid} loc [lrange $
25    geo::db($i) 1 end] }
26 }

```

Figure 7. Excerpt of generated index selection (lxworlddynamic).

The listing shows an excerpt of the generated index selection. Matrix elements are, e.g., sites. Context elements and

references are, e.g., cities, mounts, lakes, roads, rails, rivers, and grids. Any part of the dynamically generated components can be individualised depending on context-sensitive attributes and workflow configuration. The listing (Figure 8) shows an example for the generated on-the-fly-symbol used for "sites".

```

1 set bitmaps(site) [image create bitmap -data [strimj::xbm
2   "
3   ....#....
4   ...###...
5   .....#..
6   .#.###.#.
7   ##.###.##
8   .#.###.#.
9   .....#..
10  ...###...
11  ....#...."] -foreground darkviolet]

```

Figure 8. Excerpt of generated on-the-fly symbol for sites (lxworlddynamic).

Different symbols can be integrated for different sites or for different groups. The objects with their symbols are only visible if the defined level, which is handled by the level handler, is active in the interactive view.

C. Dynamical visualisation and computation

The following image (Figure 9) shows a screenshot of a resulting dynamical visualisation of items in the result matrix, in this case the resulting archaeological context sites. The generated application utilises all the features so far described with the implementation.

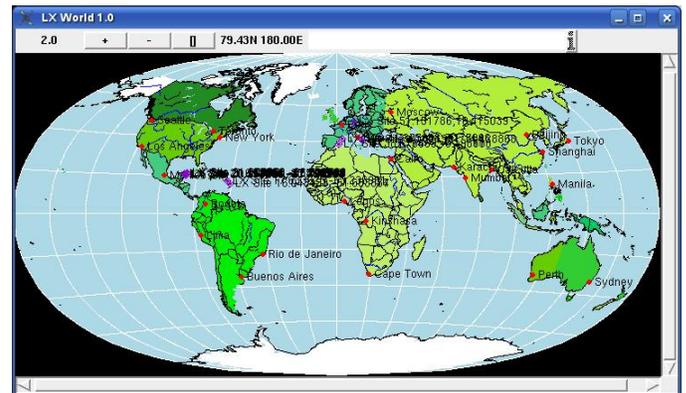


Figure 9. Archaeological context sites in interactive, dynamically generated spatial application (lxworlddynamic).

The screenshot illustrates the dynamical visualisation of the matrix elements for the context of the respective results. With the workflow a spatial context has been chosen for the matrix, creating the components. The spatial application component has been assembled by the workflow, integrating the object item references from the result matrix and secondary information from the referring knowledge resources objects with a dynamical and interactive view of the matrix.

Figure 10 shows a screenshot of a resulting dynamical zoom visualisation of matrix results and secondary information on geological and archaeological context sites. The partially

shown superpositioning effect of the respective zoom is still visible in order to show the results, which can be separated in different cognitive views, different zooms, and event sensitive actions. The implementor can do anything with this feature he is interested in, e.g., use the interactive features for label stacks and level effects, being sensitive for single results or result group. Workflow sensitive cartographic material objects (e.g., cities, land, sea, countries, border lines, grid lines) are shown for orientation and context and support cognitive feature display. The shown zoom value and the scroll bars indicate the level of detail. The screenshot illustrates the matrix elements and their references within the knowledge resources.

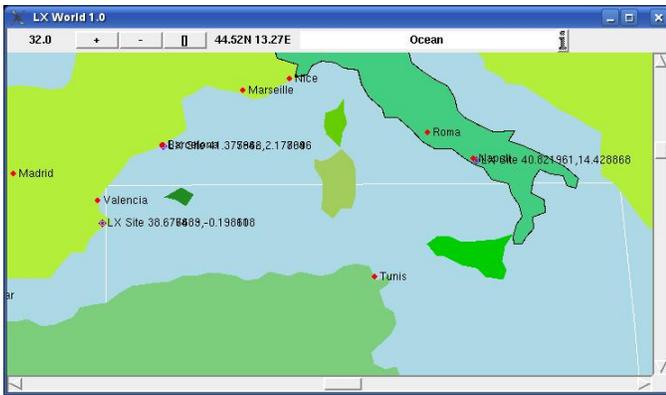


Figure 10. Zoom archaeological and geological sites and context, integrated in interactive, dynamically generated spatial application (lxworlddynamic).

In this example, an active context-sensitive window component is delivering the secondary information. This component is actively communicating with the other components. The site entity data regarding levels and objects referred from the matrix elements is dynamically available (Figure 11).

```

1 LX Site 16.043153, -61.663374
2
3 level 2
4 : site
5 loc 16.043153 61.663374
    
```

Figure 11. Example of a single site entity data extract (lxworlddynamic).

Here the matrix elements are referring to the attributes of the knowledge resources' objects, e.g., sites and cities. In this example, the displayed data excerpt includes the level, the type, and the location of the respective site. Figure 12 shows a screenshot of the corresponding dynamical site database.

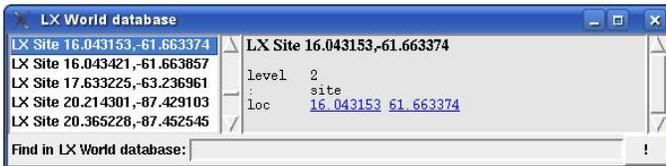


Figure 12. Site database, secondary information (lxworlddynamic).

The site database can be accessed by integrated or external applications' components, e.g., for searches, references or for

generating further result matrices. The screenshot illustrates the matrix elements and their references within the knowledge resources. In this example, an active context-sensitive window is delivering the secondary information.

X. DYNAMICAL COGNOSTIC PROCESSING

Some of the knowledge object data is processed and filtered for cognostic context and shown in application components, e.g., as secondary data in dynamical windows, e.g., in the LX World database windows.

A. Knowledge levels and cognostics

A resulting selection can, for example, be used for interactive discovery processes in the dynamically generated result matrix. The next passages are based on a result matrix of objects from the context of natural sciences and archaeology. The result matrix elements were dynamically processed and added to the interactive application component. The standard entries had not to be removed as from the processing workflow they are not interlinked and as they demonstrate the co-existence of non-linked objects. The data is based on the knowledge resources and the lxworlddynamic components. The workflow illustrates the secondary information from the dynamically extendable database being used for cognostic based improvement of an intermediate result matrix. In this case, the goal is to look for archaeological sites with associations to more than two societies or cultures as they might be called. The following example (Figure 13) shows a search window searching "Society" context in the intermediate result matrix in the intermediate historical result matrix.

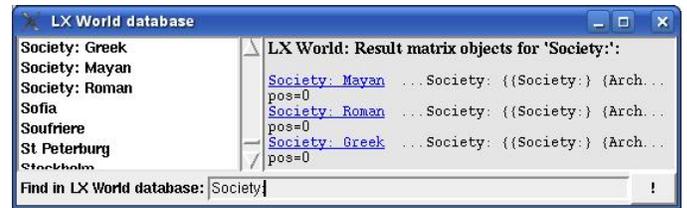


Figure 13. Secondary information on "Society" (lxworlddynamic).

The secondary search window contains the results from within the result matrix, Mayan, Roman, and Greek society. They are dynamically interlinked with deeper respective information. The following example (Figure 14) shows the referenced information for "Mayan society" in the context of the result matrix.

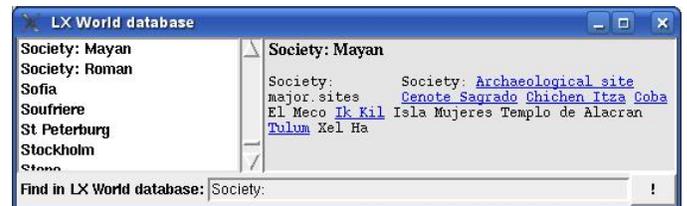


Figure 14. Dynamically linked information on "Mayan society" (lxworlddynamic).

The following example (Figure 15) shows the referenced information for “Greek society” in the result matrix’ context.

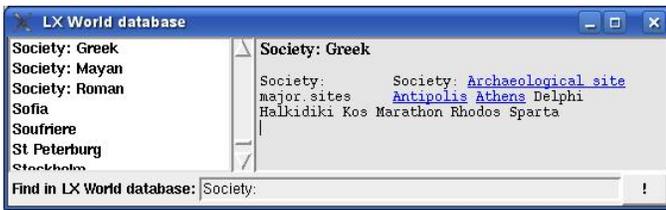


Figure 15. Dynamically linked information on “Greek society” (lxworlddynamic).

The following example (Figure 16) shows the referenced information for “Roman society” in the result matrix’ context.

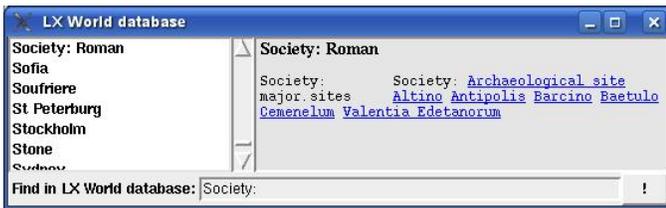


Figure 16. Dynamically linked information on “Roman society” (lxworlddynamic).

Whereas the first entry does not carry a dynamical reference to a common location, both entries Greek and Roman link to a location object “Antipolis”.

The following example (Figure 17) shows the referenced information for “Antipolis” in the context of the result matrix.



Figure 17. Dynamically linked information on “Antipolis” linked as well from “Greek Society” and “Roman society” (lxworlddynamic).

When the discovery goes into the elements’ context, then the attributes refer to many details of the context. Figure 18 shows an excerpt of an referenced Antipolis object entry with UDC classified knowledge objects.

```

1 Antipolis [Archaeology, Geophysics, Remote Sensing]:
2   Greek city, later Roman city, Southern France.
3   Modern location name Antibes, between Nice and
4   Cannes, France.
5   Predecessor of the city of Antibes, France.
6   %%IML: UDC:711(38)(37)=12=14=13
    
```

Figure 18. Content-supported cognostics: Knowledge object “Antipolis” (LX resources).

This not only refers to the antique Greece and to the antique Rome and to both cultural ‘areas’ but also to a related predecessor name of the city. Table I shows the referred conceptual knowledge.

Table I. Universal Decimal Classification location and language (knowledge resources, excerpt, English version).

UDC Code	Description
UDC:(37)	Italia. Ancient Rome and Italy
UDC:(38)	Ancient Greece
UDC:=12	Italic languages
UDC:=13	Romance languages
UDC:=14	Greek (Hellenic)

As used in the objects’ excerpt, the classification presents a view, which includes a cognostic meaning with its selection and sort order. In this case, only appropriate elements from the compiled result matrix are linked automatically when relevant for the application scenario.

B. Synonyms and cognostics

The following example (Figure 19) shows a search window searching in the intermediate result matrix for the synonyms of Vesuvius.



Figure 19. Dynamically linked information on “Vesuvius” synonyms (lxworlddynamic).

The excerpt result matrix provides quite a number of Vesuvius synonyms. Any further discovery can exploit this context for views and cognostics, either proceeding with the discovery in-width or in-depth.

Figure 20 shows an excerpt of an referenced Vesuvius object entry with UDC classified knowledge objects.

```

1 Vesuvius [Vulcanology, Geology, Archaeology]:
2   (lat.) Mons Vesuvius.
3   (ital.) Vesuvio.
4   (deutsch.) Vesuv.
5   Volcano, Gulf of Naples, Italy.
6   Complex volcano (compound volcano).
7   Stratovolcano, large cone (Gran Cono).
8   Volcano Type: Somma volcano,
9   VNUM: 0101-02=,
10  Summit Elevation: 1281\UD{m}.
11  The volcanic activity in the region is observed
12  by the Oservatorio
13  Vesuviano. The Vesuvius area has been declared a
14  national park on
15  \isodate{1995}{06}{05}. The most known antique
16  settlements at the
17  Vesuvius are Pompeji and Herculaneum.
18  Syn.: Vesaevus, Vesevus, Vesbius, Vesvius
19  s. volcano, super volcano, compound volcano
20  s. also Pompeji, Herculaneum, seismology
    compare La Soufriere, Mt. Scenery, Soufriere
    %%IML: UDC:[911.2+55]:[57+930.85]:[902]"63"
    (4+23+24)=12=14
    %%IML: GoogleMapsLocation: http://maps.google.de
    /maps?hl=de&gl=de&vpsrc=0&ie=UTF8&ll
    =40.821961,14.428868&spsn=0.018804,0.028238&t=h&
    z=15
    
```

Figure 20. Content-supported cognostics: Knowledge object “Vesuvius” (LX resources).

Whereas in this example level 1 only contains continents, some border lines, and some larger cities level 2 also visualises the targeted objects.

D. Conceptual support for generators

The high level of required application and discovery processes leads to the fact that there must be facilities to create result matrices via many different ways when using a knowledge resources concept.

Structured long-term resources integrating complex documentation and classification and providing the required 'data features' have been developed for that purpose. Besides the large spectrum of features of the knowledge objects, the major attribute supporting cognostic reuse is conceptual knowledge.

The knowledge object can carry many attributes, e.g., references, keywords, and conceptual knowledge. In many cases this conceptual knowledge comes in the form of one or more classifications.

The classification, which has shown up being especially important for complex multi-disciplinary long-term classification with knowledge resources is the Universal Decimal Classification (UDC) [77].

UDC allows an efficient and effective processing of knowledge data and provides facilities to obtain a universal and systematical view on classified objects.

Regarding library applications only, UDC is used by more than 144,000 institutions and 130 countries [78]. Further operational areas are author-side content classifications and museum collections, e.g., with documentation of resources, library content, bibliographic purposes on publications and references, for digital and realia objects.

The tiny unsorted excerpts of knowledge resources objects only refer to main UDC-based classes, which for this part of the publication are taken from the Multilingual UDC Summary (UDCC Publication No. 088) [49] released by the UDC Consortium under the Creative Commons Attribution Share Alike 3.0 license [79] (first release 2009, subsequent update 2012).

If nothing special is mentioned the basic classification codes are used in an unaltered way. If a classification refers to a modified code in particular contributing authors have to notice and document the modifications explicitly. So in practice, the classification views with knowledge resources are references to UDC.

E. Series of cognostic views

The series of screenshots in Figure 22 depicts the sequence zoom details in levels 2 to level 5 for the archaeological and geological sites and context. The first image shows the LX sites generated from the result matrix. The second image shows the zoom of the Europe LX sites' area. The third and fourth images show a zoom and a followup zoom of the Europe selection, e.g., identifying Barcin and Vesuvius sites.

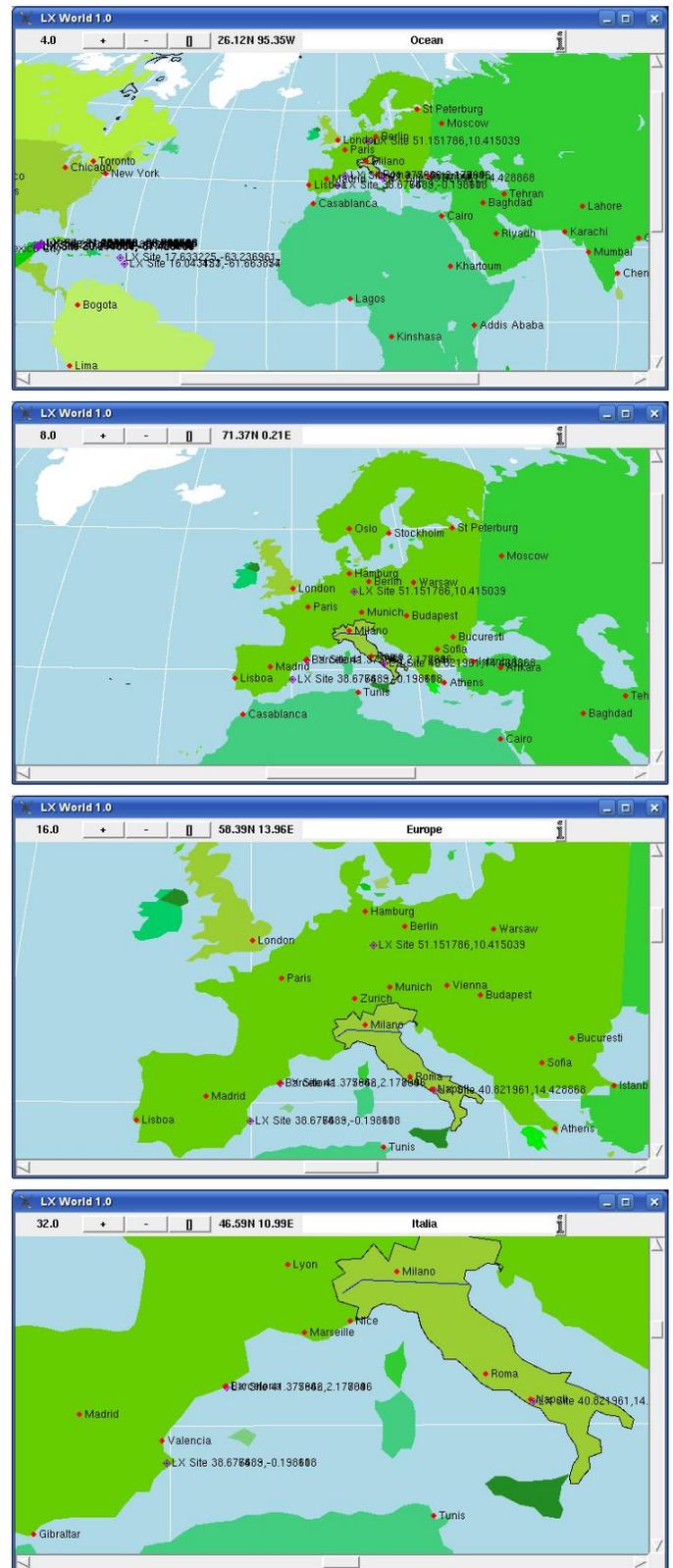


Figure 22. Ixworlddynamic – archaeological and geological sites and context, integrated in interactive, dynamically generated spatial application (top to bottom: Zoom details, level 2 – 5).

F. Triggering events and interactive features

Figure 23 shows the generated search window for the result matrix site database, especially with the Vesuvius georeference data.

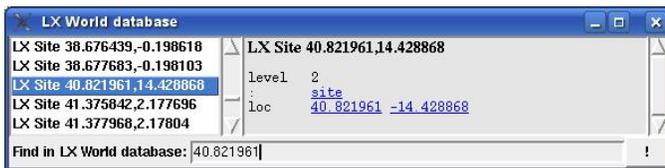


Figure 23. lxworlddynamic – archaeological and geological sites and context, site database window (Vesuvius).

Using the “!” function triggers a “goto” for the coordinates and open a corresponding zoom window containing the selected georeferences (Figure 24).

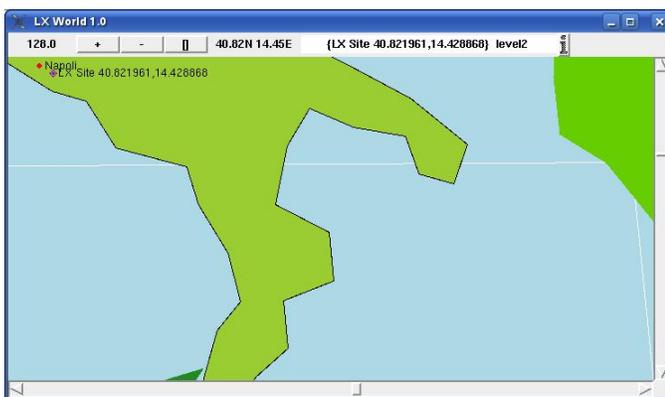


Figure 24. lxworlddynamic – archaeological and geological sites and context, “goto” (!) zoom window (Vesuvius).

The sequence of the consequent levels and interactive features shows a dynamical visualisation integrated with result matrices and components generated from universal knowledge resources. Based on this any events can be triggered that way, e.g., visualisation or computation events. This goto function is only an example for the multitude of possibilities to add code and procedures to the generated components in order to support the dynamic and cognostic features.

G. Generating referenced information code

Figure 25 shows the generated search window for the result matrix site database, especially with a country dependent graphics included with the information. The code for the component as well as for the content has been created dynamically.



Figure 25. lxworlddynamic – archaeological and geological sites and context, site database window (Coba).

The listing (Figure 26) shows an excerpt of the generated database entry code for the database information windows.

```

1 + Cimiez @ flags/fr.gif : {{Archaeological site}
  in France, country in Europe} capital Paris country.
  code FR
2 + Altinum @ flags/it.gif : {{Archaeological site}
  in Italy, country in Europe} capital Rome country.
  code IT
3 + Barcin @ flags/es.gif : {{Archaeological site}
  in Spain, country in Europe} capital Madrid country.
  code ES
4 + Coba @ flags/mx.gif : {{Archaeological site}
  in Mexico, country in America}
5 + Tulum @ flags/mx.gif : {{Archaeological site}
  in Mexico, country in America}
6 + Vesuvius @ flags/it.gif : {{Volcanological site}
  in Italy, country in Europe} capital Rome country.
  code IT
7 + Soufriere @ flags/fr.gif : {{Volcanological site}
  on Guadeloupe, France, Caribbean, F.W.I.}
8 + {Mt. Scenery} @ flags/nl.gif : {{Volcanological site}
  on Saba, The Netherlands, Caribbean, D.W.I.}

```

Figure 26. Excerpt of generated database entry code for the database information windows (lxworlddynamic).

The excerpt includes the references for the hypertext and flag code for parts of the result matrix elements. The database code can be used for many purposes, as codes generated like this part are capable of being directly integrated into the dynamical application components as shown above. The text is automatically linked by the framework environment in a very simple fuzzy way and can transport cognostic aspects in textual and visual context. The cognostic levels can be handled very flexible, e.g., the content and the generated database entry code in arbitrary ways. In this case, the result matrix defines objects for one level. The environment defines additional context objects for other levels.

XI. EVALUATION

Application components can be created and assembled dynamically from any workflow. Knowledge objects can be used efficiently with any dynamical components. The cognostic features for the content as well as for resulting application components can be efficiently provided by conceptual knowledge documented with advanced knowledge resources. Database interfaces can be used dynamically and efficiently with the components. Graphical User Interfaces (GUI) and Application Programming Interfaces (API) can be used most flexible on that dynamical base.

The information created from arbitrary numbers of resources’ objects in this excerpt includes site labels, level information, category information, as well as the georeferences. The components can trigger any instances and events dynamically and interactively. This allows any kind of processing, computation, and visualisation, from sketch like visualisation to special cartographical mapping. All the more, the dynamical components based on the knowledge resources and the lxworlddynamic frameworks allow for the interactive, autonomous components’ generation. The components have been successfully implemented on a number of operating systems (e.g., SuSE Linux, Debian, Red Hat, and Scientific Linux, as well with older and up-to-date distributions).

The knowledge and system architecture allows to seamlessly integrate with all the steps required for a sustainable implementation of the methodological bases. The case studies done over several decades of knowledge resources' creation and development and two decades of application component developments have shown that plans for extending structures, creating interfaces, self-learning components, container formats, and integration with life cycles and systems resources' operation can be assured even for long-term application.

XII. FURTHER POTENTIAL OF COMPONENTS

The development with this long-term research has led to sustainable solutions when implementing components that support splitting up the data, which is most important in order to achieve an efficient use of available resources, e.g., Input/Output or distribution onto many disks and machines. Otherwise, bulk data can be as inefficient to handle as it will not be practical to use certain methods and algorithms at all. Not all the tools, which are implemented by respective disciplines for handling data are suitable and efficient for all data, algorithms, and workflows. Parallel processing can and should be done with different tools for different purposes. Examples are commonly used tools like Hadoop, Hadoop Distributed File System (HDFS), and MapReduce with a basic Application Programmers Interface (API) for a more abstracted way of developing components for parallel processing of portions of data on a cluster system.

For critical cases it might be argued that components like MapReduce code are costly to write and difficult to troubleshoot and manage. With the same breath, tools like MapReduce are inefficient for exploring datasets. Therefore, the Hadoop_on_a_chip 'ecosystem' got tools that span a wide variety of needs, including providing layers of abstraction that make interacting with the data simpler.

In public presentation and marketing common understanding of containers is reduced to certain aspects, e.g., security features [80], [81]. Special container concepts have been introduced for handling Big Data, especially scientific data, e.g., the NERSC 'Shifter' at the National Energy Research Scientific Computing Center [82].

These approaches are insufficient from the content and knowledge related point of view of containers. For example, on the one hand, there is nothing general for a container when postulating that it should contain "everything". At the same level, security features have to be considered very special with specific cases of application. In addition, in most cases the advanced application scenarios trigger those secondary conditions. On the other hand, containers must not only provide computational features. The result is that up to now we neither have a commonly discussed container concept nor a set of universal features. A container is a term for a data or file format bundling the data for certain purpose and application. The features for anything more interesting will include data, documentation, references and so on. Doing so includes how the information is transferred or accessed, which includes to define the modalities for a certain scenario. When discussing containers from the content and knowledge view

many scenarios if not universal application is the target, which induces a much more general understanding of containers as has been done with this research.

XIII. CONCLUSIONS

Application components have shown to benefit in many cases from dynamical processing in context with knowledge and cognostics. This research has provided details of the implementation and the resulting components, from structure, content, and context of knowledge resources and cognostic support for dynamical components, geo sets, computation to index selection and various views from the resulting visualisations.

Long-term knowledge resources are core means for providing a sustainable base for these features. The knowledge resources can refer to any kind of resources and allow to efficiently transport implementation features for long-term vitality of use cases. Over many years the implementation with different scenarios has proven to be straight forward.

It has been demonstrated that with the proposed framework and concept context-sensitive dynamical components can be successfully created on base of universal knowledge resources. It has been shown that even standalone dynamical components can be created based on the implementation of the foundations, supporting arbitrary dynamical, modular, portable, and extendable database application programming and database graphical user interfaces. The implementation can utilise workflows and algorithms for knowledge discovery and selection up to intelligent application component creation. It integrates very efficiently in workflow chains, e.g., for computation and visualisation, and is very well usable even for rapid prototyping environments.

The integration is non-invasive regarding the knowledge resources for uni-directional visualisation and computation. If the intention with an application scenario is to update information consistently then multi-directional workflows can also update objects in side knowledge resources or containers from the created components in arbitrary ways, ensuring consistency and plausibility, as well as following management and security policies. Future work will be focussed on issues and usability beyond the plain Big Data approaches, resulting from data values, creating implementations supporting data vitality. A major focus will concentrate on sustainable integration of content and context with conceptual knowledge and cognostics.

ACKNOWLEDGEMENTS

We are grateful to all national and international partners in the GEXI cooperations for the innovative constructive work. We thank the Science and High Performance Supercomputing Centre (SHPSC) for long-term support of collaborative research since 1997, including the GEXI developments and case studies. We are grateful to the "Knowledge in Motion" (KiM) long-term project and its scientific members, DIMF, for the contributions on the development and application of classifications, especially to Dr. Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek (GWLb) Hannover, to Dipl.-Biol. Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, and

to Dipl.-Ing. Martin Hofmeister, Hannover, for practical multi-disciplinary case studies and the analysis of advanced concepts. Many thanks go to the Institute for Legal Informatics (IRI), Leibniz Universität Hannover, and to the scientific colleagues at the Westfälische Wilhelms-Universität (WWU), for discussion, support, and sharing experiences on collaborative computing and knowledge resources and for participating in fruitful case studies, as well as to the participants of the postgraduate European Legal Informatics Study Programme (EULISP) for prolific discussion of scientific, legal, and technical aspects over the last years. We are grateful to Dipl.-Ing. Hans-Günther Müller, Cray, for previous work on flexible practical solutions to architectural challenges and excellent technical support. Last but not least we thank Richard Suchenwirth for his initial implementation of Tclworld, which could be used as a tool for handling result matrices.

REFERENCES

- [1] C.-P. Rückemann, "Creating Knowledge-based Dynamical Visualisation and Computation," in Proceedings of The Seventh International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2015), February 22 – 27, 2015, Lisbon, Portugal. XPS Press, 2015, pages 56–62, ISSN: 2308-393X, ISBN-13: 978-1-61208-383-4, ISBN-13: 978-1-61208-048-2 (CDROM), URL: http://www.thinkmind.org/download.php?articleid=geoprocessing_2015_3_40_30063 [accessed: 2015-04-12].
- [2] "Europeana," 2015, URL: <http://www.europeana.eu/> [accessed: 2015-04-12].
- [3] "WDL, World Digital Library," 2015, URL: <http://www.wdl.org> [accessed: 2015-04-12].
- [4] F. Hülsmann and C.-P. Rückemann, "Funding Value and Data Value," KiM Summit, June 15, 2015, Knowledge in Motion, "Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF)", Hannover, Germany, 2015.
- [5] UDC, Universal Decimal Classification. British Standards Institute (BSI), 2005, complete Edition, ISBN: 0-580-45482-7, Vol. 1 and 2.
- [6] "UDC Online," 2015, <http://www.udc-hub.com/> [accessed: 2015-04-12].
- [7] "Universelle Dezimalklassifikation (UDK)," 2014, Uni-Protokolle, URL: http://www.uni-protokolle.de/Lexikon/Universelle_Dezimalklassifikation.html [accessed: 2015-04-12].
- [8] C.-P. Rückemann, "Decision Making and High End Computing," International Tutorial, October 21, 2012, The Second International Conference on Advanced Communications and Computation (INFOCOMP 2012), International Conference on Advances in Information Mining and Management (IMMM 2012), International Conference on Social Eco-Informatics (SOTICS 2012), International Conference on Mobile Services, Resources, and Users (MOBILITY 2012), International Conference on Global Health Challenges (GLOBAL HEALTH 2012), International Conference on Communications, Computation, Networks and Technologies (INNOV 2012) / DataSys 2012, October 21–26, 2012, Venice, Italy, [Lecture], 2012, URL: http://www.aria.org/conferences2012/filesINFOCOMP12/rueckemann_infocomp2012_tutorial_sec.pdf.
- [9] C.-P. Rückemann, F. Hülsmann, and M. Hofmeister, "Cognostics and Applications for Knowledge Resources and Big Data Computing," KiM Summit, November 9, 2015, Knowledge in Motion, "Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF)", Hannover, Germany, 2015.
- [10] J. W. Tukey and P. A. Tukey, "Computer Graphics and Exploratory Data Analysis: An Introduction," in Proceedings of the Sixth Annual Conference and Exposition of the National Computer Graphics Association, Dallas, Texas, USA, April 14–18, 1985. Fairfax, VA: National Computer Graphics Association, 1985, pp. 772–785, reprinted in W. S. Cleveland (ed.), The Collected Works of John W. Tukey, Vol 5 (pp. 419–436), New York: Chapman & Hall.
- [11] W. S. Cleveland, Dynamic Graphics for Statistics. Wadsworth & Brooks/Cole, Statistic/Probability Series, 1988, McGill, M. E. (ed.), ISBN: 0-534-09144-X.
- [12] T. L. Nyerges and D. M. Mark, Cognitive aspects of human-computer interaction for geographic information systems. Kluwer Academic Publishers, 1995.
- [13] M. F. Goodchild, Modelling error in objects and fields. Taylor & Francis, London, 1989, in: The Accuracy of Spatial Databases, Goodchild, M. F. and Gopal, S. (eds.), ISBN: 0-85066-847-6.
- [14] B. Tversky, "Cognitive Maps, Cognitive Collages, and Spatial Mental Models," Lecture Notes in Computer Science, vol. 716, 1993, pp. 14–24, Plenum Press, New York.
- [15] D. R. Montello, "Scale and Multiple Psychologies of Space," in Spatial Information Theory: a theoretical basis for GIS, ser. Lecture Notes in Computer Science, A. Frank and C. I., Eds., no. 716, COSIT'93, Berlin: Springer Verlag, 1993, pp. 312–321.
- [16] G. Edwards, "The Voronoi Model and Cultural Space: Applications to the Social Sciences and Humanities," in Spatial Information Theory: A Theoretical Basis for GIS, ser. Lecture Notes in Computer Science, A. U. Frank and I. Campari, Eds., vol. 716. Berlin: Springer Verlag, 1993, pp. 202–214.
- [17] G. Edwards, "Geocognostics – A New Paradigm for Spatial Information?" in Proceedings of the AAAI Spring Symposium 1996, 1996.
- [18] S. Hirtle, "Towards a Cognitive GIS, Proceedings of the Advanced Geographic Data Modelling Workshop," Journal of the Netherlands Geodetic Commission, vol. 40, 1994, pp. 217–227.
- [19] C.-P. Rückemann, "Dynamical Parallel Applications on Distributed and HPC Systems," International Journal on Advances in Software, vol. 2, no. 2, 2009, ISSN: 1942-2628, URL: <http://www.ariajournals.org/software/>, [accessed: 2009-11-16].
- [20] C.-P. Rückemann, "Integrating Future High End Computing and Information Systems Using a Collaboration Framework Respecting Implementation, Legal Issues, and Security," International Journal on Advances in Security, vol. 3, no. 3&4, 2010, pp. 91–103, savola, R., (ed.), pages 91–103, ISSN: 1942-2636, LCCN: 2008212457 (Library of Congress), URL: http://www.ariajournals.org/security/sec_v3_n34_2010_paged.pdf [accessed: 2015-06-07].
- [21] C.-P. Rückemann, Integrated Information and Computing Systems for Advanced Cognition with Natural Sciences. Premier Reference Source, Information Science Reference, IGI Global, 701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA, Oct. 2012, pp. 1–26, chapter I, in: Rückemann, C.-P. (ed.), Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences, 543 (xxiv + 519) pages, 21 chapters, ill., ISBN-13: 978-1-4666-2190-9 (hardcover), EISBN: 978-1-4666-2191-6 (e-book), ISBN: 978-1-4666-2192-3 (print & perpetual access), DOI: 10.4018/978-1-4666-2190-9, LCCN: 2012019278 (Library of Congress), OCLC: 798809710, URL: <http://www.igi-global.com/chapter/integrated-information-computing-systems-advanced/70601> [accessed: 2015-08-02], DOI: <http://dx.doi.org/10.4018/978-1-4666-2190-9.ch001> [accessed: 2015-08-02].
- [22] C.-P. Rückemann, "From Multi-disciplinary Knowledge Objects to Universal Knowledge Dimensions: Creating Computational Views," International Journal On Advances in Intelligent Systems, vol. 7, no. 3&4, 2014, pp. 385–401, Bodendorf, F., (ed.), ISSN: 1942-2679, LCCN: 2008212456 (Library of Congress), URL: http://www.ariajournals.org/intelligent_systems/intsys_v7_n34_2014_paged.pdf [accessed: 2015-06-07].
- [23] "Wissensarbeiter erfinden das Rad immer wieder neu," library essentials, LE Informationsdienst, Mai/Juni 2015, 2015, pp. 30–32, ISSN: 2194-0126, URL: <http://www.libess.de> [accessed: 2015-06-07].
- [24] "The Knowledge Rework Report," 2015, December 2014, URL: <http://www.knowledge-rework.com/>

- pages.coveo.com/KM-RESEARCH-SURVEY-RESULTS-LP.html [accessed: 2015-06-07].
- [25] "Industrie 4.0: Realität oder Wunschvorstellung?" library essentials, LE-Informationdienst, Mai/Juni 2015, 2015, pp. 27–29, ISSN: 2194-0126, URL: <http://www.libess.de> [accessed: 2015-06-07].
- [26] "Jedem dritten Produktionsbetrieb ist Industrie 4.0 kein Begriff," 2015, Pressemitteilung vom 07. April 2015, URL: http://www.bitkom.org/de/presse/8477_81970.aspx [accessed: 2015-06-07].
- [27] "4 von 10 Unternehmen nutzen Industrie 4.0-Anwendungen," 2015, Pressemitteilung vom 13. April 2015, URL: http://www.bitkom.org/de/presse/8477_82018.aspx [accessed: 2015-06-07].
- [28] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch, "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries," 2015, Boston Consulting Group - BCG, April 2015, URL: http://www.bcgperspectives.com/Images/Industry_40_Future_of_Productivity_April_2015_tcm80-185183.pdf [accessed: 2015-06-07].
- [29] "Neue Bevölkerungsvorausberechnung für Deutschland bis 2060," 2015, Pressemitteilung Nr. 153 vom 28.04.2015, URL: https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2015/04/PD15_153_12421.html [accessed: 2015-06-07].
- [30] "Die Roboter kommen: Folgen der Automatisierung für den deutschen Arbeitsmarkt," 2015, Studie vom 30. April 2015, URL: https://www.ing-diba.de/imperia/md/content/pw/content/ueber_uns/presse/pdf/ing_diba_economic_research_die_roboter_kommen.pdf [accessed: 2015-06-07].
- [31] H. Lau, "Four Ways Your Data is Lying to You," Datanami, 2015, June 8, 2015, URL: <http://www.datanami.com/2015/06/08/four-ways-your-data-is-lying-to-you/> [accessed: 2015-06-16].
- [32] A. Woodie, "Big Data's Dirty Little Secret," Datanami, 2015, July 2, 2015, URL: <http://www.datanami.com/2015/07/02/big-datas-dirty-little-secret/> [accessed: 2015-07-05].
- [33] A. Gray, "Big Data's Small Lie – The Limitation of Sampling and Approximation in Big Data Analysis," Datanami, 2015, July 20, 2015, URL: <http://www.datanami.com/2015/07/20/big-datas-small-lie-the-limitation-of-sampling-and-approximation-in-big-data-analysis> [accessed: 2015-07-26].
- [34] C. Lukasik, "Big Data, Big Misnomer," Datanami, 2015, July 22, 2015, URL: <http://www.datanami.com/2015/07/22/big-data-big-misnomer> [accessed: 2015-07-26].
- [35] B. Gersbeck-Schierholz and C.-P. Rückemann, "All-in Folders: Challenges and Benefits," KiMrise, Knowledge in Motion, June 20, 2015, Circle Summit Workgroup Meeting, "Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF)", Düsseldorf, Germany, 2015.
- [36] "LX-Project," 2015, URL: <http://www.user.uni-hannover.de/cpr/x/rprojs/en/#LX> (Information) [accessed: 2015-04-12].
- [37] E. Dodsworth and L. W. Laliberte, Eds., *Discovering and using historical geographic resources on the Web: A practical guide for Librarians*. Lanham: Rowman and Littlefield, 2014, ISBN: 0-8108-914-1.
- [38] S. A. Keller and R. Schneider, Eds., *Wissensorganisation und -repräsentation mit digitalen Technologien*. Walter de Gruyter GmbH, 2014, Bibliotheks- und Informationspraxis, ISSN: 0179-0986, Band 55, ISBN: 3-11-031270-0.
- [39] E. W. De Luca and I. Dahlberg, "Die Multilingual Lexical Linked Data Cloud: Eine mögliche Zugangsoptimierung?" *Information Wissenschaft & Praxis*, vol. 65, no. 4–5, 2014, pp. 279–287, Deutsche Gesellschaft für Information und Wissen e.V. (DGI), Ed., De Gruyter Saur, ISSN: 1434-4653, e-ISSN: 1619-4292, DOI: 10.1515/iwp-2014-0040, (title in English: The Multilingual Lexical Linked Data Cloud: A possible semantic-based access to the Web?).
- [40] L. Schumann and W. G. Stock, "Ein umfassendes ganzheitliches Modell für Evaluation und Akzeptanzanalysen von Informationsdiensten: Das Information Service Evaluation (ISE) Modell," *Information Wissenschaft & Praxis*, vol. 65, no. 4–5, 2014, pp. 239–246, Deutsche Gesellschaft für Information und Wissen e.V. (DGI), Ed., De Gruyter Saur, ISSN: 1434-4653, e-ISSN: 1619-4292, DOI: 10.1515/iwp-2014-0043, (title in English: A comprehensive holistic model for evaluation and acceptance analyses of information services: The Information Service Evaluation (ISE) model).
- [41] G. Isaew and A. Roganow, "Qualitätssteuerung von Informationssystemen: Theoretisch-methodologische Grundlagen," *Information Wissenschaft & Praxis*, vol. 65, no. 4–5, 2014, pp. 271–278, Deutsche Gesellschaft für Information und Wissen e.V. (DGI), Ed., De Gruyter Saur, ISSN: 1434-4653, e-ISSN: 1619-4292, DOI: 10.1515/iwp-2014-0044, (title in English: Quality Management of Information Systems: Theoretical and methodological basics).
- [42] "Geo Exploration and Information (GEXI)," 1996, 1999, 2010, 2015, URL: <http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI> [accessed: 2015-04-12].
- [43] C.-P. Rückemann, "Active Map Software," 2001, 2005, 2012, URL: <http://wwwmath.uni-muenster.de/cs/u/ruckema> (original permalink) [accessed: 2012-01-01], URL: <http://www.user.uni-hannover.de/cpr/x/publ/2001/dissertation/wwwmath.uni-muenster.de/cs/u/ruckema> (information, data, abstract) [accessed: 2015-04-18].
- [44] C.-P. Rückemann, "Knowledge Integration for Scientific Classification and Computation," in *The Fourth Symposium on Advanced Computation and Information in Natural and Applied Sciences, Proceedings of The 12th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM)*, September 22–28, 2014, Rhodes, Greece, Proceedings of the American Institute of Physics (AIP), AIP Conference Proceedings, no. 1. AIP Press, American Institute of Physics, Melville, New York, USA, 2014, ISSN: 0094-243X, ISBN-13: 978-0-7354-1287-3, DOI: 10.1063/1.4912358.
- [45] C.-P. Rückemann, "Long-term Sustainable Knowledge Classification with Scientific Computing: The Multi-disciplinary View on Natural Sciences and Humanities," *International Journal on Advances in Software*, vol. 7, no. 1&2, 2014, pp. 302–317, ISSN: 1942-2628, LCCN: 2008212462 (Library of Congress), URL: http://www.ariajournals.org/software/soft_v7_n12_2014_paged.pdf [accessed: 2015-06-07].
- [46] "GMT - Generic Mapping Tools," 2015, URL: <http://imima.soest.hawaii.edu/gmt> [accessed: 2015-04-12].
- [47] C.-P. Rückemann, "High End Computing Using Advanced Archaeology and Geoscience Objects," *International Journal On Advances in Intelligent Systems*, vol. 6, no. 3&4, 2013, pp. 235–255, ISSN: 1942-2679, LCCN: 2008212456 (Library of Congress), URL: http://www.ariajournals.org/intelligent_systems/intsys_v6_n34_2013_paged.pdf [accessed: 2015-04-12].
- [48] C.-P. Rückemann, "Enabling Dynamical Use of Integrated Systems and Scientific Supercomputing Resources for Archaeological Information Systems," in *Proc. INFOCOMP 2012*, Oct. 21–26, 2012, Venice, Italy, 2012, pp. 36–41, ISBN: 978-1-61208-226-4, URL: http://www.thinkmind.org/download.php?articleid=infocomp_2012_3_10_10012 [accessed: 2015-04-12].
- [49] "Multilingual Universal Decimal Classification Summary," 2012, UDC Consortium, 2012, Web resource, v. 1.1. The Hague: UDC Consortium (UDCC Publication No. 088), URL: <http://www.udcc.org/udccsummary/php/index.php> [accessed: 2015-04-12].
- [50] F. Cremer, C. Engelhardt, and H. Neuroth, "Embedded Data Manager – Integriertes Forschungsdatenmanagement: Praxis, Perspektiven und Potentiale," *Bibliothek: Forschung und Praxis*, vol. 39, no. 1 (April), 2015, pp. 13–31, ISSN: 0341-4183, DOI: <http://dx.doi.org/10.1515/bfp-2015-0006> [accessed: 2015-05-10].
- [51] "Museum-Digital," 2015, URL: <http://www.museum-digital.de/> [accessed: 2015-05-17].
- [52] G. Wurzer, K. Kowarik, and H. Reschreiter, Eds., *Agent-based Modeling and Simulation in Archaeology*. Springer, 2015, ISBN: 978-3-319-00008-4, URL: <http://www.springer.com/978-3-319-00007-7> [accessed: 2015-05-17].
- [53] X. Rubio-Campillo, *Large Simulations and Small Societies: High Per-*

- formance Computing for Archaeological Simulations. Springer, 2015, in: Agent-based Modeling and Simulation in Archaeology, Wurzer, G., Kowarik, K., and Reschreiter, H. (eds.), ISBN: 978-3-319-00008-4, URL: <http://www.springer.com/978-3-319-00007-7> [accessed: 2015-05-17].
- [54] "Geschlossene Systeme als Bedrohung für das freie Web," library essentials, LE_Informationdienst, Mai/Juni 2015, 2015, pp. 24–26, ISSN: 2194-0126, URL: <http://www.libess.de> [accessed: 2015-06-07].
- [55] J. Boie, "Schleichender Tod der Webseite," 2015, in: Süddeutsche Zeitung Online, 28. April 2015, URL: <http://www.sueddeutsche.de/digital/ende-des-world-wide-web-das-netz-wird-neu-geknuepft-1.2454673> [accessed: 2015-06-07].
- [56] "The Domain Name Industry Brief," 2015, Vol. 12, No. 1, March 2015, URL: <http://www.verisigninc.com/assets/domain-name-report-march2015.pdf> [accessed: 2015-06-07].
- [57] T. Dunning, "The Role of Bias In Big Data: A Slippery Slope," Datanami, 2015, August 3, 2015, URL: <http://www.datanami.com/2015/08/03/the-role-of-bias-in-big-data-a-slippery-slope/> [accessed: 2015-08-09].
- [58] T. Trader, "Reading List: Fault Tolerance Techniques for HPC," HPCwire, 2015, August 6, 2015, URL: <http://www.hpcwire.com/2015/08/06/reading-list-fault-tolerance-techniques-for-hpc/> [accessed: 2015-08-15].
- [59] B. Gersbeck-Schierholz and C.-P. Rückemann, "Coping with Failures in Information and Applied Knowledge," KiMrise, Knowledge in Motion, June 20, 2015, Circle Summit Workgroup Meeting, "Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF)", Brussels-Welckenrath, Belgium, 2015.
- [60] C.-P. Rückemann, "Archaeological and Geoscientific Objects used with Integrated Systems and Scientific Supercomputing Resources," International Journal on Advances in Systems and Measurements, vol. 6, no. 1&2, 2013, pp. 200–213, ISSN: 1942-261x, LCCN: 2008212470 (Library of Congress), URL: http://www.thinkmind.org/download.php?articleid=sysmea_v6_n12_2013_15 [accessed: 2015-04-12].
- [61] "Tcl Developer Site," 2015, URL: <http://dev.scriptics.com/> [accessed: 2015-04-12].
- [62] "Open-MPI," 2015, URL: <http://www.open-mpi.org> [accessed: 2015-04-12].
- [63] R. Suchenwirth, "Tclworld, Version 0.6, Konstanz, 2002-01-31," 2002, URL: http://wiki.tcl.tk/_repo/wiki_images/tm06.zip [accessed: 2015-04-12].
- [64] R. Suchenwirth, "dbapi – A simple database API," 2002, URL: http://wiki.tcl.tk/_repo/wiki_images/tm06.zip [accessed: 2015-04-12].
- [65] R. Suchenwirth, "dbgui.tcl – A little database GUI," 2002, URL: http://wiki.tcl.tk/_repo/wiki_images/tm06.zip [accessed: 2015-04-12].
- [66] W. Gaul and M. Schader, Eds., Classification As a Tool of Research. North-Holland, Amsterdam, 1986, Proceedings, Annual Meeting of the Classification Society, (Proceedings der Fachtagung der Gesellschaft für Klassifikation), ISBN-13: 978-0444879806, ISBN-10: 0-444-87980-3, Hardcover, XIII, 502 p., May 1, 1986.
- [67] P. Cousson, "UDC as a non-disciplinary classification system for a high-school library," in Proceedings UDC Seminar 2009, Classification at a Crossroads: Multiple Directions to Usability, 1992, pp. 243–252, URL: http://www.academia.edu/1022257/UDC_as_a_non-disciplinary_classification_system_for_a_high-school_library [accessed: 2015-04-12].
- [68] A. Adewale, "Universal Decimal Classification (UDC): A Most For All Libraries," library 2.0, the future of libraries in the digital age, 2014, URL: <http://www.library20.com/forum/topics/universal-decimal-classification-udc-a-most-for-all-libraries> [accessed: 2015-04-12].
- [69] J. Templin and L. Bradshaw, "Measuring the Reliability of Diagnostic Classification Model Examinee Estimates," Journal of Classification, vol. 30, no. 2, 2013, pp. 251–275, Heiser, W. J. (ed.), ISSN: 0176-4268 (print), ISSN: 1432-1343 (electronic), URL: <http://dx.doi.org/10.1007/s00357-013-9129-4> [accessed: 2015-04-12].
- [70] P. Cerchiello and P. Giudici, "Non parametric statistical models for on-line text classification," Advances in Data Analysis and Classification – Theory, Methods, and Applications in Data Science, vol. 6, no. 4, 2012, pp. 277–288, special issue on "Data analysis and classification in marketing" Baier, D. and Decker, R. (guest eds.) ISSN: 1862-5347 (print), ISSN: 1862-5355 (electronic).
- [71] F. Heel, "Abbildungen zwischen der Dewey-Dezimalklassifikation (DDC), der Regensburger Verbundklassifikation (RVK) und der Schlagwortnormdatei (SWD) für die Recherche in heterogen erschlossenen Datenbeständen – Möglichkeiten und Problembereiche," Bachelorarbeit im Studiengang Bibliotheks- und Informationsmanagement, Fakultät Information und Kommunikation, Hochschule der Medien Stuttgart, 2007, Bearbeitungszeitraum: 1. Juni 2007 bis 31. August 2007, Stuttgart, August 2007, URL: http://opus.bs-z-bw.de/hdms/volltexte/2009/665/pdf/BA_Fabian_Heel.pdf [accessed: 2015-04-12].
- [72] "Physics and Astronomy Classification Scheme, PACS 2010 Regular Edition," 2010, American Institute of Physics (AIP), URL: <http://www.aip.org/pacs> [accessed: 2015-04-12].
- [73] "Mathematics Subject Classification (MSC2010)," 2010, URL: <http://msc2010.org> [accessed: 2015-04-12].
- [74] R. S. Sasser, U.S. Geological Survey library classification system. U.S. G.P.O., 1992, USGS Bulletin: 2010.
- [75] "North American Industry Classification System (NAICS), Concordances," 2014, URL: <https://www.census.gov/eos/www/naics/concordances/concordances.html> [accessed: 2015-04-12].
- [76] F. Hülsmann and C.-P. Rückemann, "Intermediate Classification and Concordances in Practice," KiMrise, Knowledge in Motion Meeting, January 9, 2015, Knowledge in Motion, "Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF)", Hannover, Germany, 2015.
- [77] "Universal Decimal Classification Consortium (UDCC)," 2015, URL: <http://www.udcc.org> [accessed: 2015-04-12].
- [78] A. Slavic, "UDC libraries in the world - 2012 study," universaldecimalclassification.blogspot.de, 2012, Monday, 20 August 2012, URL: <http://universaldecimalclassification.blogspot.de/2012/08/udc-libraries-in-world-2012-study.html> [accessed: 2015-04-12].
- [79] "Creative Commons Attribution Share Alike 3.0 license," 2012, URL: <http://creativecommons.org/licenses/by-sa/3.0/> [accessed: 2015-04-12].
- [80] G. Leopold, "Can Containers Contain?" Remains Top Security Issue," Enterprisetech, 2015, July 29, 2015, URL: <http://www.enterprisetech.com/2015/07/29/can-containers-contain-remains-top-security-issue/> [accessed: 2015-08-02].
- [81] G. Leopold, "Goldman Sachs Brokers Container Spec Deal," Enterprisetech, 2015, June 22, 2015, URL: <http://www.enterprisetech.com/2015/06/22/goldman-sachs-brokers-container-spec-deal/> [accessed: 2015-08-02].
- [82] K. Kincade, "NERSC's 'Shifter' Makes Container-based HPC a Breeze," HPCwire, 2015, August 7, 2015, URL: <http://www.hpcwire.com/2015/08/07/nerscs-shifter-makes-container-based-hpc-a-breeze/> [accessed: 2015-08-15].

Qualitative Comparison of Geocoding Systems using OpenStreetMap Data

Konstantin Clemens

Service-centric Networking
Telekom Innovation Laboratories
Technische Universität Berlin, Germany
konstantin.clemens@campus.tu-berlin.de

Abstract—*OpenStreetMap* is a platform where users contribute geographic data. To serve multiple use cases, these data are held in a very generic format. This makes processing and indexing *OpenStreetMap* data a challenge. *Nominatim* is an open source geocoding system that consumes *OpenStreetMap* data. *Nominatim* processes *OpenStreetMap* data well. It relies on predefined address schemes to determine the meaning of various address elements and to discover relevant results. *Nominatim* ranks results by a global precomputed score. *Elasticsearch* is a web service on top of *Lucene* – a general purpose document store. *Lucene* searches for documents and ranks results according to a *term frequency – inversed document frequency* scoring scheme. In this article, *Nominatim* is compared to two systems populated with exactly the same data: An out-of-the-box instance of *Elasticsearch*, and a specialized system that builds on top of *Elasticsearch*, but implements a custom algorithm to aggregate house numbers on every street segment, thereby vastly reducing the index size. The three geocoding systems are thoroughly benchmarked with three different data sets and geocoding queries of increasing complexity. The analysis shows: *Term frequency – inversed document frequency* based ranking yields more accurate results, and is more robust removing the need for predefined address schemes. Also, the reduced index size of the specialized system comes at a cost, which, depending on the application scenario, may be a viable option.

Keywords—*Geocoding; Address Indexing; OpenStreetMap; Nominatim; Elasticsearch*

I. INTRODUCTION

This article is based on [1], which showed that generic document stores as *Elasticsearch* make a good baseline for geocoding services. A more detailed view on the experiments is presented here, additional experiments are introduced, as well as a new, more complex data set. Also, a specialized approach to reduce the index size of the document store is suggested in this article. It is evaluated in the same way as the two systems that have

already been presented in [1].

OpenStreetMap [2] is a collaborative platform where every registered user can contribute relevant and mappable data. These data can be of arbitrary type: Street segments or bridges, their names, speed limits or surface materials, position and outline of buildings, forests, lakes or administrative areas, radio beacons and their operators, train tracks, bicycle, hiking or bus routes, etc. That variety is possible, because *OpenStreetMap* data is stored in a very generic data format without a specific application in mind.

OpenStreetMap data consist of the three entity types *node*, *way*, and *relation*. Every entity type has an *id* attribute for referencing and the attributes *timestamp*, *version*, and *changeset* for versioning. There are also the attributes *uid* and *user* specifying the user who created or modified an entity.

Nodes have values for *longitude* and *latitude* in WGS84 format [3]. Therefore, nodes model single points on the globe. Because most things on a map have an areal extent, a node alone can be used to specify the position of an entry with yet unknown area. Also, for example, nodes can specify mountain peaks, magnetic and geographic poles, or other entities with no area. Most of the nodes, however, are parts of ways and relations.

Ways compose lines through points by specifying ordered lists of node references. A way can also specify an area by referencing the same node at the begin and at the end of the node list. Using ways it is possible to model car lanes on a street, pedestrian zones, simple outlines of structures, and the like.

Relations, in turn, may reference both ways and nodes. Therefore, relations can model complex geographic features as polygons with holes as well as specify, e.g., a center point for displaying pins on the map. Relations may also reference other relations assembling abstract entities that span several 'real things' such as

groups of islands, or universities with multiple, wide-spread buildings.

Finally, nodes, ways, and relationships can hold an arbitrary number of *tags*. Tags are key-value pairs that specify names, categories, address elements as city, district or street name, house number ranges, data sources, speed limits, and all other attributes of real-world features that the entities model.

Clearly, this data format is flexible enough to accommodate all kinds of data. For example, properly tagged relationships could hold 3D models of buildings, where different parts of buildings are different relations. [4] describes in depth how 3D models are stored in OpenStreetMap data. On one hand, that flexibility is convenient, but on the other hand it is also an obstacle: For many use cases the data need to be preprocessed before they can be utilized. In the case of geocoding, because of the versatile structure of OpenStreetMap data, address elements are often spread across different entities. For example, a node might only be tagged with a house number, while the way that references this node only holds the street name information. The way itself may be contained in a postal code area represented by a relation. Another relation could describe the area of the city. However, the relations not necessarily reference the way or each other. Therefore, to offer a geocoding service, addresses need to be assembled out of OpenStreetMap entities first.

Geocoding [5] is the process of resolving named locations such as full addresses, named areas, and sometimes even landmarks into their location. A variety of proprietary and open source geocoding services, e.g., [6], [7], or [8] offer that through their APIs. First, to offer such a service, there need to be source data to be made searchable. These data need to contain a mapping from addresses or names to the geolocation of the entry, which, most often, is represented as a pair of WGS84 coordinates. Because errors in data are exposed through the service, the quality of data determines the quality of the geocoding service. To some extent, errors in the data can be covered when the data is being indexed: To account for errors and ambiguities in queries indexing algorithms need to store the data in a way that it is fuzzy and robust at the same time – some of these techniques can be applied to the source data at indexing time too. Geocoding services use indexes to parse and split queries into address elements, possibly determine their meaning and compile a list of candidate results the query may have referred to. In a last step, the service orders the

candidate list so that the most probable result is on top. If possible, unlikely candidates at the bottom of that list are cut off. Depending on the specific type of geocoder, specialized approaches are used as described in [9], [10], and [11].

Nominatim [7] is an open source system implemented in several programming languages that uses OpenStreetMap data to provide a geocoding service. That means Nominatim preprocesses the OpenStreetMap data assembling full addresses from address elements spread in tags on various entities prior to building an index for geocoding. Both these processes are very time consuming, also because Nominatim precomputes global ranks for all entries at indexing time already. The source data, the geocoding index, and the ranks are all stored in a PostGIS [12] enabled PostgreSQL [13] data base. An apache server [14] invoking PHP to access the data base serves search requests via HTTP. When a search is performed, Nominatim first parses the query address according to predefined schemes. These schemes specify where various address elements as postal code, state, city, district, street name, or house number may be located in a query. Next, Nominatim queries the data base using the derived address elements. It collects candidate results ordered by the precomputed ranking score. In Nominatim there is no stage where result lists are ordered or cut in relation to a served query.

Term frequency – inverted document frequency (TF-IDF) [15], [16] is a formula to rank documents based on query terms and their distributions. For a given query term and a document, the term frequency is the number of occurrences of that term in that document. A higher term frequency implies that a document is more relevant to a query: The more often a query term appears in a document, the more likely the document corresponds to the query. At the same time, if a term rarely occurs in a document, the term may have been mentioned as a side note only, while the actual topic of the document could be a different one. Some terms are very common or have multiple meanings. Such terms appear in many documents therefore. Other terms are rare and specific. Clearly, rare and specific terms distinguish the relevant documents stronger than the generic and ambiguous ones. To incorporate that, a global term weight is computed for each term: The document frequency of a term is the number of documents that term occurs in. The more documents a term occurs in, the less distinguishable that term is. Thus, for each term a greater inverted document frequency implies a greater importance of its

term frequency value. The TF-IDF score for a term and a document is therefore computed by multiplying the term frequency and the inverted document frequency. For a query with multiple terms, the overall TF-IDF score is computed as the sum of scores of every query term for each document. There are variants of TF-IDF that differ in various details. Particularly, BM25f [17] is a variant that supports documents with differently weighted fields. Also, when computing the BM25f score, the document length is taken into account.

In contrast to Nominatim, *Lucene* [18] is a generic open source document indexing framework, that ranks results at query time. Lucene supports various ranking schemes for ordering results, including TF-IDF and BM25f. In the context of geocoding, where addresses are documents, one can easily agree that the token 'street' is less distinctive than the token 'Springfield', which – given that there are many towns called like that – is less distinctive than, let's say 'Chicago'. Therefore, for a query for 'Michigan Street Chicago' the 'Michigan Avenue in Chicago, Illinois' is scored higher than the 'Michigan Street in Springfield, Massachusetts'. The match on the rare token 'Chicago' outweighs the match on the common token 'Street'.

Elasticsearch [19] is a RESTful [20] wrapper around Lucene. That means, besides many other features it offers a simple HTTP based interface to create, read, update, and delete single documents as well as whole document collections. Internally a separate Lucene index is maintained per document collection, allowing Elasticsearch to expose a search interface as well. Thus, if populated with documents that contain addresses and their coordinates, Elasticsearch becomes an HTTP based geocoding service that uses BM25f to compute, which results match to given queries best.

A number of efforts was made to index OpenStreetMap data in Elasticsearch. The *elasticsearch-osmosis-plugin* [21] extends *Osmosis* [22], a tool for processing OpenStreetMap data, allowing it to index this data in Elasticsearch. Thereby, the entities are indexed as they are, the plug-in does not transform the data in any way. Instead it enables Elasticsearch to be used as a tool to browse original OpenStreetMap entities from the indexed source data set. *Pelias* [23] is a collection of modular tools that plug into each other. There are modules for reading data from OpenStreetMaps as well as other data formats, a module for indexing the data in Elasticsearch, and a module to offer a set of APIs and provide a geocoding service on top. Similar

to the *elasticsearch-osmosis-plugin*, *Pelias* does not process the data in any way similar to Nominatim. Both systems do not harvest addresses that are spread across OpenStreetMap entities, but rather index the raw data in its generic format. *Gazetteer* [24] consists of two modules: One to parse OpenStreetMap data and derive points of interest, addresses, streets, street networks, and administrative boundaries. The other to index this data in Elasticsearch and thereby offer a geocoding service. Unlike the *osmosis-elasticsearch-plugin* or *Pelias*, the *Gazetteer* tool tries to assemble full addresses based on entities that contain one another, or are located nearby. Nominatim, as discussed earlier, is a geocoding service that does not rely on Elasticsearch, Lucene, or a dynamic ranking scheme. Instead, Nominatim tries to split queries into address elements in accordance to given schemes and ranks candidates by a globally precomputed score. As the *Gazetteer*, Nominatim is collecting address elements from various nodes as it is necessary with the generic OpenStreetMap data format.

For all the systems above, no qualitative analysis on their performance is available. These systems are best-effort solutions. In this article, a set of experiments is performed, which allows a qualitative comparison of geocoding systems based on BM25f as used in generic document stores and Nominatim, which relies on address schemes to split queries into address elements and derive their meaning. Existing solutions are using different data than Nominatim: They either do not preprocess OpenStreetMap data at all, or, like the *Gazetteer* solution, use own processing. Therefore, for this article, the data indexed in Elasticsearch is extracted directly from a Nominatim data base. This ensures that there are no data differences between the measured systems. Only differences between the various indexing algorithms are evaluated. In addition to Nominatim and Elasticsearch, an approach to reduce the index size is suggested in this article: Instead of indexing every house number address as a separate document, house numbers on the same street segment are aggregated into one document. Instead of WGS84 coordinates each document in this system contains a mapping table declaring the coordinates for every house number on that street segment. This solution requires a thin layer around Elasticsearch, which takes care of identifying and, if available, extracting the response for a requested house number. Elasticsearch is still used to retrieve street segments that match to queries. These three solutions are evaluated using the same benchmarks allowing to derive the strengths and

placex		
ID;TEXT;LATLON;PARENT_ID	ID: 5 TEXT: Berlin LATLON: 52.5075419,13.4251364	ID: 5 TEXT: Berlin LATLON: 52.5075419,13.4251364
5; Berlin; 52.5075419,13.4251364; _	ID: 10 TEXT: 10587 Berlin LATLON: 52.5182401,13.3172961	ID: 10 TEXT: 10587 Berlin LATLON: 52.5182401,13.3172961
10; 10587; 52.5182401,13.3172961; 5	ID: 20 TEXT: Ernst Reuter Platz, 10587 Berlin LATLON: 52.5127183,13.3217624	ID: 20 TEXT: Ernst Reuter Platz, 10587 Berlin LATLON: 52.5127183,13.3217624
20; Ernst Reuter Platz; 52.5127183,13.3217624, 10	ID:200 TEXT:Ernst Reuter Platz 7, 10587 Berlin LATLON: 52.5127183,13.3217624	ID: 20 TEXT: Ernst Reuter Platz, 10587 Berlin LATLON: 52.5127183,13.3217624 HOUSE NUMBERS: - 7 ID:200 TEXT:Ernst Reuter Platz 7, 10587 Berlin LATLON: 52.5127183,13.3217624
200; 7; 52.5128751,13.3203904; 20	ID:201 TEXT:Ernst Reuter Platz 9, 10587 Berlin LATLON: 52.5129359,13.3203243	- 9 ID:201 TEXT:Ernst Reuter Platz 9, 10587 Berlin LATLON: 52.5129359,13.3203243
201; 9; 52.5129359,13.3203243; 20		

Figure 1. Schemes of indices in Nominatim (left, simplified), Elasticsearch (middle), and EAHN (right)

weaknesses of every system. All in all, BM25f strives to bring the document to the top of the result lists that, according to term hits and their respective weights, matches to a given query the best. Also, [25] shows, that address schemes often contradict each other, e.g., by some schemes assuming the house number before the street name while other schemes assume it behind the street name, which results in ambiguous parsing of address elements. Therefore, it is fair to assume that the Elasticsearch based solutions are more solid geocoding services than Nominatim.

II. EXPERIMENT

Three geocoding systems: Nominatim, Elasticsearch, and *Elasticsearch with Aggregated House Numbers* (EAHN) have been set up with the same data in their indexes. First, Nominatim has been set up with OpenStreetMap data for Europe. This long-running process is fully automated and results in a PostgreSQL data base that holds the data and the indexes for Nominatim. The entries served by Nominatim are stored in a single table, which is the output of processing OpenStreetMap data and assembling address elements spread across entities. Besides a name, e.g., the name of a street or the house number, rows of this table store their areal extent in PostGIS geometries, their IDs, types, ranks, and some additional meta-data. Additionally, each row references the rows containing the data of the next-higher administrative area: A row with a house number references the row with the street name the house number belongs to, which in turn references the row with the city district that street segment is in, and so forth. This allows entries to be normalized, not storing the names of higher level administrative areas that make up a full address. The full address hierarchy is assembled by a stored procedure, which is called by Nominatim for each result at query time.

This same stored procedure and a PostGIS procedure to derive the centroid latitude and longitude of PostGIS geometries were applied to every row of Nominatim's result table generating documents to be indexed in Elasticsearch. Documents derived this way contain the full address of an entry (which, in the case of higher level administrative areas, may not contain the most specific address parts), the entries' ID in the Nominatim data base, and WGS84 latitude and longitude coordinates. An Elasticsearch instance was set up next to Nominatim and populated with the documents extracted.

One drawback of indexing every address in Elasticsearch is the vast amount of denormalized data: Street names, districts, cities, postal codes, etc. do not differ for many addresses, but are all stored and indexed as parts of separate documents for every house number address. This requires additional space and makes the process of picking the right document harder: The BM25f scores of different documents are less spread apart if the documents scored contain redundant data. EAHN tries to approach this problem, by only creating separate documents for separate street segments. Thus, in EAHN BM25f is only used to find the right street segment. All house numbers of that street segment are stored within that document. They are, however, not indexed in Elasticsearch and extracted using string matching in a second step.

To set up EAHN with the same data, the same data base table was used. First, as for documents in Elasticsearch, for every row the full address for its entry as well as the entries' ID and WGS84 coordinates have been assembled. Entries representing a house number address were separated from non house number entries using a flag Nominatim stores in the data base as part of each row's meta data. House number entries were grouped by the ID of their parent, which, according to

Nominatim's data base structure, would be the street segment the house numbers would be in. Non house number entries were used to generate documents to be indexed in EAHN. In a final step, documents of entries referenced by house number entries were extended to contain a mapping from each house number to the respective house number address, its ID and its WGS84 coordinates. These documents with aggregated house numbers as well as all documents that were not referenced by a house number entry were indexed in an Elasticsearch instance. Note that by default Elasticsearch would index every part of the document. To achieve the reduction in index size, the instance used for EAHN was explicitly configured to index the address text of street segments only, ignoring house number mappings and their addresses.

Figure 1 presents the simplified schemes of the three systems. While Nominatim uses a normalized table with entries referencing their parents, Elasticsearch contains a denormalized document per entry. EAHN aggregates documents representing house numbers in their parent document, containing a denormalized document for each entry that is not a house number. Looking at the schemas in Figure 1 makes it obvious that Elasticsearch is a geocoding system already: A response to a search request contains addresses with their WGS84 coordinates that match to a given query. That is not the case for EAHN: It requires an additional step between its Elasticsearch instance and a client that unwraps documents representing house numbers if appropriate. This layer has been implemented as follows:

- 1) Forward every query to Elasticsearch, request it to highlight those parts of the query that have matched to parts of the document.
- 2) For every candidate in the returned list, for every query token that has not been highlighted as matched, if the token is a key in the house number mapping, return the text, the ID, and the WGS84 coordinates the key maps to.
- 3) Fall back to returning the text, the ID, and the WGS84 coordinates of the first candidate on the result list if no suitable house number result has been found.

For EAHN, ideally, a street segment would be the entire part of a street referenced by the same address, i.e., the segment has the same city, the same district, the same postal code, etc. However, OpenStreetMaps defines street segments on a finer-grained level: Multiple equally addressed street segments exist next to each other. These segments represent different sides or multiple chunks

of a lengthy street, or parts of a street that vary in other attributes as speed limits. Some house numbers would be attached to one street segment, while others would be attached to another. For this article, the street segments, as they are defined in OpenStreetMaps were used. Therefore, it is not sufficient for EAHN to only look at the house numbers attached to the first candidate returned by Elasticsearch. Without house numbers being part of the scoring scheme BM25f cannot differentiate between otherwise equal documents in a meaningful way. Thus, it is inevitable to look at all equally named street segments. For experiments in this article, EAHN was configured to query Elasticsearch for 250 candidates to look at – a number small enough to be processed and big enough to cover almost all of the cases with multiple equally named street segments. On the downside, requesting many candidates for a street that does not have many segments exposes the risk to match a house number on a segment of a different street than the one the query referred to. To account for that, a house number match has only be returned as stated in (2) if the number of query tokens matching the candidate was equal or greater than the number of matched tokens in the first candidate. Thereby, every matched token was counted only once, even if the indexed document contained that token multiple times. That was required to deal with documents that stated address elements multiple times. Such documents appeared as artifacts of Nominatim collecting address elements from incorrectly tagged OpenStreetMap entities. For example, entities for city districts often list the postal codes contained in those districts repeatedly. Another caveat of EAHN are house number ranges. Since there is no standardized way to specify them, various separators are used between the leading and the tailing number in OpenStreetMap data as well as in the data sets used for experiments. Because house number ranges were not normalized by EAHN, some times, a requested house number range would not be retrieved, even though it was available in the data. Similarly, letters added to house numbers were not normalized and not found if the index had a different capitalization or whitespace variant of the same house number. Generally, because house numbers have not been indexed in the documents, they did not contribute to the BM25f score of the document leaving a greater chance of false documents appearing on top of the result list.

Given all these downsides of EAHN, its main feature only becomes obvious after it has been populated with data: Because instead of every house number address

TABLE I. Index sizes

system	row or document count	index size on disk
Nominatim	61 198 211	(366 GB)
Elasticsearch	61 198 211	15.7 GB
EAHN	33 678 783	6.2 GB

being a separate document house numbers are aggregated into the document of their street segment, the number of indexed documents is drastically reduced. Also, house numbers themselves are not indexed in EAHN at all. Therefore, while all of the data is still available the index size is much smaller too. Table I shows the index sizes of the three systems benchmarked in this article after they have been populated with OpenStreetMap data for Europe. Nominatim collected and indexed 61.2M addresses from OpenStreetMap data. Every single entry in Nominatim is indexed as a separate document in Elasticsearch. Aggregating the house numbers on the same street segment into one document nearly cuts the number of documents to be indexed in EAHN in half. Because the house number tokens are not indexed any more, and because overall there are less documents to reference, the index size of EAHN shrinks by more than 50%. Note that the index size of Nominatim is not comparable with the sizes of the other two indexes. The given size of the PostgreSQL data base after Nominatim's processing of OpenStreetMap data also includes the source entities as well as preliminary results of the processing. To operate Nominatim as a geocoder, many of these tables are not required. Also, Nominatim contained and indexed the full set of address translations, which have not been exported or indexed. This feature has not been evaluated on Elasticsearch and EAHN, though it is fair to assume that address translations can be handled in the same way as addresses in their local languages.

As a final difference between Nominatim and the other two systems, abbreviations are to be mentioned. Nominatim implements a defined set of operations to, e.g., normalize STREET to ST in both the indexed data and a served query. These operations have not been ported to Elasticsearch or EAHN. Because Nominatim returned addresses are not abbreviated, the documents indexed in Elasticsearch and EAHN did not contain any abbreviations implicitly. To mimic the feature, test sets were manually processed to spell out every abbreviated address element.

At the end of the set up process three geocoding

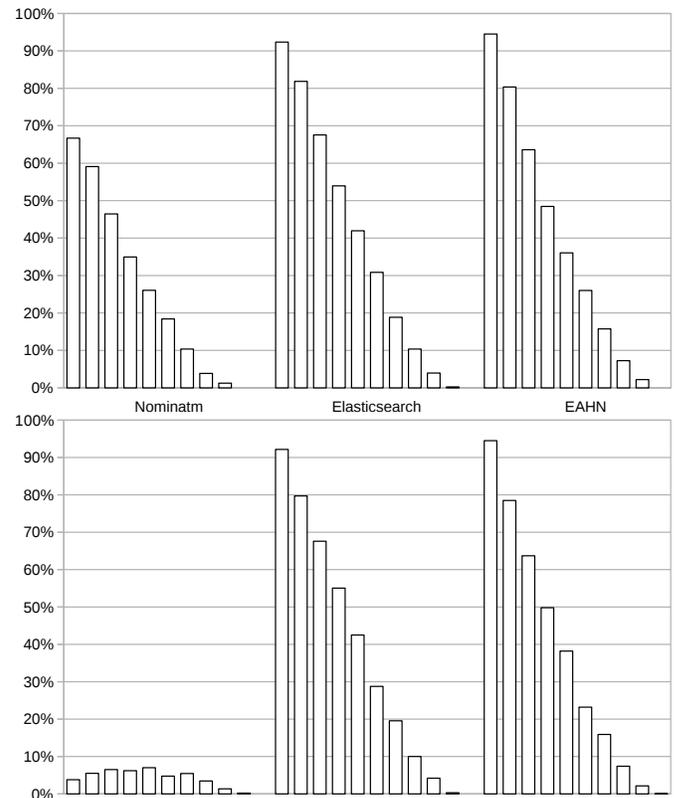


Figure 2. Percentages of successfully geocoded addresses with increasingly less address tokens for **addresses extracted from Nominatim** with address tokens in order (top) and shuffled (bottom)

systems with the same addresses indexed were running next to each other: Nominatim with precomputed rank scores as well as BM25f based Elasticsearch and EAHN. To compare these three systems three benchmarks have been designed to gradually increase the complexity of requests issued. The benchmarks allow to observe how the percentage of successfully served requests decreases for the three systems. Three data sets have been generated for the three systems:

- 1) A data set of 2000 addresses randomly sampled from the Nominatim data base.
- 2) A data set of 1500 addresses of pharmacies in big German cities.
- 3) A data set of 2000 addresses randomly sampled out of addresses sourced from a German on-line portal for real estate.

In accordance to the set up, set (1) of addresses extracted from Nominatim (and, therefore, indexed in each system) contain samples from various European countries. The addresses were extracted with their IDs so that right or wrong responses from the geocoding systems

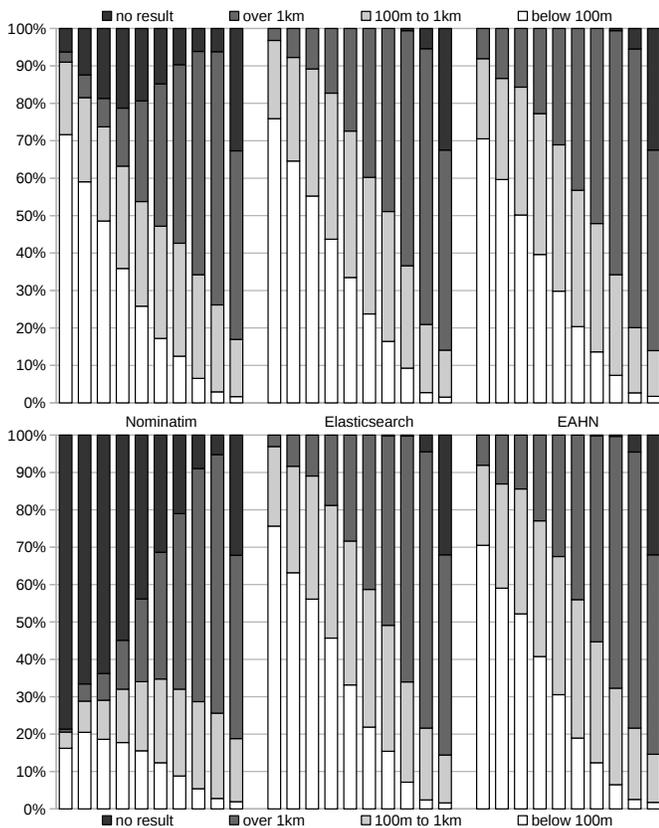


Figure 3. Percentages of buckets of geocoding responses to requests with **formally correct postal addresses** and increasingly less address tokens with address tokens in their original order (top) and shuffled (bottom)

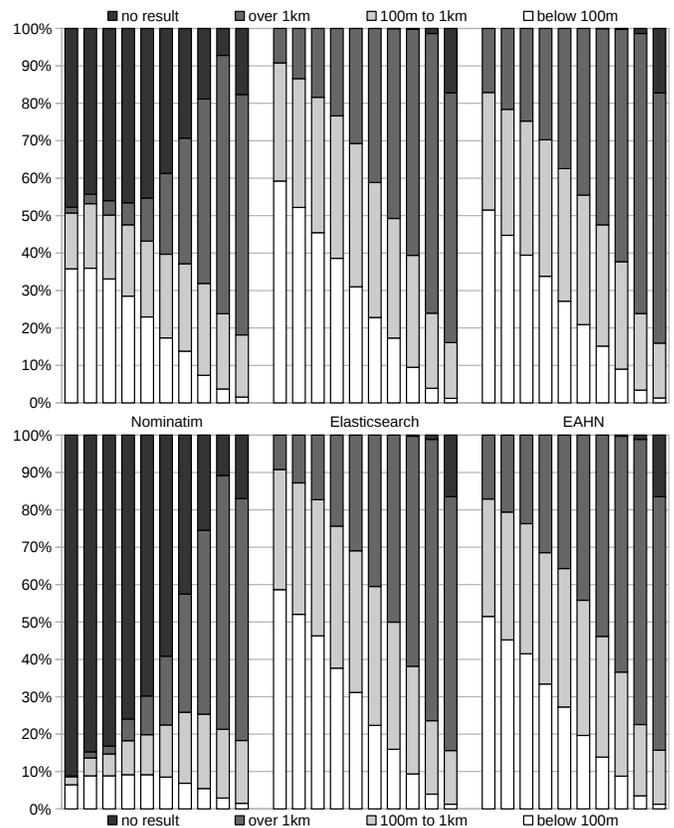


Figure 4. Percentages of buckets of geocoding responses to requests with **user input addresses** and increasingly less address tokens with address tokens in their original order (top) and shuffled (bottom)

were identified by simple ID comparison. The addresses were extracted exactly as Nominatim presents them to clients. This presentation is usually verbose: It names additional address elements as district or county names that are not usually part of a postal address. In contrast to that, set (2) contains formally correct postal addresses of pharmacies. Note that for some of the pharmacies OpenStreetMap data (and, therefore, the three geocoding systems) contain the addresses as well as the pharmacies them selves as separate entries. This does not interfere with the approach used to evaluate if a request with this address has been successful or not: The addresses have been geocoded with the Google’s geocoding system as reference first. Because Google’s geocoding system uses different data than Nominatim, Elasticsearch, and EAHN, results were not expected have equal coordinates. Instead results were grouped into buckets depending on the distance to the reference. The buckets *within 100m*, *100m-1000m*, *further than 1000m*, and *no result* have been used. Because of these distance based buckets, it did not matter if a geocoding system returned the result

for an address or the pharmacy itself: The positions of both entries are close enough so that their distance to the reference position would up in the same bucket for either of the two cases. For a general purpose geocoder (and, therefore, in this article) only results from the first bucket are close enough to be considered as successfully served ones. Set (3) contains yet again a different type of addresses: The on-line portal is asking agents to input addresses of the real estate they are offering. Thus, as actual real estate is offered, all the addresses are most likely to exist and be comprehensible for human beings. More importantly, however, these addresses are not necessarily formally correct, as addresses in (2). Instead, addresses are spelled out as humans refer to them when communicating with each other. The on-line portal is also asking their users to specify geocoordinates of addresses by clicking on a map. As these WGS84 coordinates are part of the data set, relying on a reference geocoding system is not necessary. As for pharmacies, results of the real estate addresses were grouped into the same four buckets, depending on their distance to the user

specified coordinates. Again, only results in the *within 100m* bucket can be considered correct.

For every data set multiple experiments have been conducted: First, full addresses as they are in the data sets have been issued as geocoding requests to each system. Next, in steps of 10% more and more randomly picked address tokens have been removed from requests, up until requests contained only 10% of all address tokens. The consequence of dropping tokens from the query are less precise and more ambiguous requests. This behavior mimicked users that query for incomplete addresses. Finally, the whole iteration of increasingly ambiguous queries has been repeated with randomly shuffled address tokens. This behavior mimicked users that do not adhere to a formal address standard. In total for each of the 5500 addresses 10 queries with different token counts were issued with tokens in their original and in shuffled order to each of the three systems under test. Note that all queries were composed from house number addresses. A benchmark examining a production geocoding system would incorporate a contingent of queries for named areas that is proportional to the number of queries for named areas the system has to serve. In this article, however, no productive system specifies the portion of queries for named addresses. Also, it is fair to assume that geocoding house number addresses is a more complex scenario.

III. RESULTS

Measurement results for all three systems and all three data sets are presented in Figures 2, 3, and 4. Every chart has three blocks of bars – one for the reference geocoding system Nominatim, one for the document store Elasticsearch, and one for Elasticsearch with aggregated house numbers. The leftmost bar of each block shows the success rate for issuing geocoding requests with all address tokens. Every next bar shows results for requests with additional 10% address tokens dropped. Each block consists of exactly ten bars with the rightmost bar showing the success rate for geocoding requests with only 10% of address tokens. In each figure the upper chart is showing the results of stating requests with address tokens in their original order, while the bottom chart is showing the results of geocoding queries with shuffled address tokens. Note that for ordered and shuffled queries, different query tokens were dropped at random: A query for an address with address tokens in their original order would therefore contain different tokens than a shuffled query for the same address. This statistical noise explains, e.g., why in Figure 2 EAHN

seems to perform slightly better with 60% of address tokens shuffled, rather than in their original order. An insight in some specific numbers is given by Table II. It lists the percentages of successful requests for all systems and data sets with requests containing 100% and 50% of address tokens. Every value pair gives the success rate for requests with address tokens in their original order first, followed by the success rate for requests with shuffled address tokens.

Results for geocoding addresses extracted from the index are presented in Figure 2. The upper chart shows that Elasticsearch and EAHN are outperforming Nominatim with full addresses already. With less and less address tokens in the queries and, therefore, with less distinctive queries, the performance of all the three systems decreases linearly, keeping Nominatim below the two BM25f based systems constantly. Another picture is shown by the lower chart: While Elasticsearch and EAHN are almost not affected by shuffling of address tokens at all, Nominatim's success rate drops to below 5% for full addresses. With increasingly less tokens, Elasticsearch and EAHN behave as if the address tokens were in order. In contrast, Nominatim seems to become better at first, reaching its maximum at requests composed with 60% of address tokens. Even at this maximum Nominatim still stays below its own performance with address tokens in order. Interestingly, EAHN performs better than Elasticsearch for full addresses only. For addresses with missing tokens, Elasticsearch takes the lead.

Figure 3 shows response buckets of requests with formally correct postal addresses of pharmacies. The bars for the various buckets are shaded differently and stacked adding up to 100% of requests. The general trend on this figure is similar to that shown on Figure 2: Nominatim does not handle shuffled address tokens well; Elasticsearch and EAHN constantly outperform Nominatim. A closer look, however, reveals that Nominatim is more successful in dealing with formally correct postal addresses than with queries for addresses extracted from the index. Table II highlights that for the two BM25f based systems the opposite is true: On indexed addresses these systems perform best. Another thing to note on Figure 3 is that Nominatim often returns no results, while the other two systems return some responses for most of the queries. That behavior can be good or bad, depending on the use case scenario.

Results of the experiment using user input addresses in Figure 4 confirm the previous observations. Again,

TABLE II. Select rates of successfully served geocoding requests (ordered - shuffled)

	indexed addresses	formally correct postal addresses	user input addresses
100% address tokens in query			
Nominatim	66.7% - 3.8%	71.6% - 16.2%	35.8% - 6.4%
Elasticsearch	92.3% - 92.1%	75.9% - 75.6%	59.2% - 58.6%
EHN	94.5% - 94.5%	70.5% - 70.5%	51.5% - 51.5%
50% address tokens in query			
Nominatim	18.4% - 4.7%	17.2% - 12.3%	17.35% - 8.5%
Elasticsearch	30.8% - 28.7%	23.7% - 21.9%	22.8% - 22.3%
EHN	26.0% - 23.2%	20.3% - 18.9%	20.9% - 19.6%

Nominatim is constantly outperformed by the other two systems. Again, only Nominatim is impacted negatively if address tokens are shuffled. Clearly, the user input data set was the hardest to geocode: All three systems are least successful with geocoding these data.

IV. CONCLUSION

The experiments show that Nominatim requires queries to adhere to the implemented addressing schemes to function well: It performs best on the data set of formally correct postal addresses. However, if query tokens are in arbitrary order, Nominatim is likely to not find the right result. This is a strong limitation, given that there is not one universal, but many different and some times contradicting address schemes in use. Because there are less ways to shuffle less address tokens, Nominatim's performance at first increases when less tokens are used to issue a query. Addresses with missing tokens are less distinctive, which is why Nominatim does not reach the same performance as with query tokens left in their order. For the same reason the performance of the BM25f based systems decreases linearly proportional to the number of address tokens dropped. The experiments prove that BM25f is an approach suitable to select proper geocoding results. It is independent of the token order and generally performs better than precomputing global ranks. The vast superiority of the two BM25f systems on indexed addresses is somewhat artificial: Querying a system that matches documents to queries using queries generated from documents indexed is very likely to produce a high success rate. The only reason why Elasticsearch does not achieve full 100% hits is data: As discussed, some entries are repeated in the OpenStreetMaps data set, thus a duplicate to the right result has a different ID and is therefore not recognized as a correct response. This is

also the reason why EAHN outperforms Elasticsearch for full addresses: By design EAHN looks into multiple candidates for a house number match, which makes it more likely to derive the correct result. Obviously, the actual performance of a geocoding system highly depends on the actual queries it faces. When developing geocoding systems, however, Elasticsearch makes a good base line to compete with.

EAHN has proven to be a valuable approach to offer geocoding services too. As expected, it yields slightly less accurate results than Elasticsearch, because less parts of the address are indexed. Still, the observed impact has shown to be a tolerable cost, especially taking into account the gains EAHN offers: Smaller index size and less indexed documents promote responsiveness and scalability of a system. Also, a smaller index is easier to update. Future work could assess how to incorporate house numbers into the indexing and scoring methods to close the performance gap to Elasticsearch. Also, aggregating equally named street segments would lead to further reduction of the index size. Ideally, an approach can be found that utilizes the hierarchical structure of addresses on other levels as well. For example, by aggregating districts of a city in the same way house numbers of a street have been aggregated for EAHN would allow to store index data in a more compact and efficient way. Finally, to further improve EAHN, normalization logic for house number ranges could be developed. For example, house number ranges to be indexed could be unfolded at indexing time mapping every house number in the range to the entire range it is part of. At query time it would suffice to use only the starting or only the ending house number of a range to retrieve the entire range address.

REFERENCES

Advanced Geographic Information Systems, Applications, and Services, 2013, pp. 155–160.

- [1] K. Clemens, “Geocoding with openstreetmap data,” *GEOProcessing 2015*, 2015, pp. 1–2.
- [2] “OpenStreetMap,” <http://wiki.openstreetmap.org>, Nov. 2015.
- [3] National Imagery and Mapping Agency, “Department of Defense, World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems,” in *Technical Report 8350.2 Third Edition*, 2004.
- [4] M. Goetz and A. Zipf, “Openstreetmap in 3D—detailed insights on the current situation in Germany,” in *Proceedings of the AG-ILE 2012 International Conference on Geographic Information Science*, Avignon, France, 2012.
- [5] D. Goldberg, J. Wilson, and C. Knoblock, “From Text to Geographic Coordinates: The Current State of Geocoding,” *URISA Journal*, vol. 19, no. 1, 2007, pp. 33–46.
- [6] “Google Geocoding API,” <https://developers.google.com/maps/>, Nov. 2015.
- [7] “Nomatim,” <http://wiki.openstreetmap.org/wiki/Nominatim>, Nov. 2015.
- [8] “GeoNames,” <http://www.geonames.org>, Nov. 2015.
- [9] Open Geospatial Consortium, “Gazetteer Service – Application Profile of the Web Feature Service Best Practice,” Feb. 2012.
- [10] D. Yang, L. Bilaver, O. Hayes, and R. Goerge, “Improving geocoding practices: evaluation of geocoding tools,” *Journal of Medical Systems*, vol. 28, no. 4, 2004, pp. 361–370.
- [11] L. Can, Z. Qian, M. Xiaofeng, and L. Wenyin, “Postal address detection from web documents,” in *Web Information Retrieval and Integration, 2005. WIRI’05. Proceedings. International Workshop on Challenges in*. IEEE, 2005, pp. 40–45.
- [12] “PostGIS,” <http://postgis.net/>, Nov. 2015.
- [13] “PostgreSQL,” <http://www.postgresql.org/>, Nov. 2015.
- [14] “Apache HTTP Server,” <http://httpd.apache.org/>, Nov. 2015.
- [15] G. Salton and C.-S. Yang, “On the specification of term values in automatic indexing,” *Journal of documentation*, vol. 29, no. 4, 1973, pp. 351–372.
- [16] G. Salton, C.-S. Yang, and C. T. Yu, “A theory of term importance in automatic text analysis,” *Journal of the American society for Information Science*, vol. 26, no. 1, 1975, pp. 33–44.
- [17] S. Robertson, H. Zaragoza, and M. Taylor, “Simple BM25 extension to multiple weighted fields,” in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 2004, pp. 42–49.
- [18] “Lucene,” <http://lucene.apache.org/>, Nov. 2015.
- [19] “Elasticsearch,” <https://www.elastic.co/>, Nov. 2015.
- [20] A. Rodriguez, “Restful web services: The basics,” *IBM developerWorks*, 2008.
- [21] “elasticsearch-osmosis-plugin,” <https://github.com/ncolomer/elasticsearch-osmosis-plugin/wiki/Quick-start>, Nov. 2015.
- [22] “Osmosis,” <http://wiki.openstreetmap.org/wiki/Osmosis>, Nov. 2015.
- [23] “Pelias,” <https://github.com/pelias/>, Nov. 2015.
- [24] “gazetteer,” <https://github.com/kiselev-dv/gazetteer/>, Nov. 2015.
- [25] K. Clemens, “Automated processing of postal addresses,” in *GEOProcessing 2013, The Fifth International Conference on*

Conditions Necessary for Visibility and Sensors Placement in Urban Environments Using Genetic Algorithms

Oren Gal, Yerach Doytsher

Mapping and Geo-information Engineering
Technion - Israel Institute of Technology
Haifa, Israel
e-mail: {orengal,doytsher}@technion.ac.il

Abstract—Optimized coverage using multi-sensors is a challenging task, which is becoming more and more complicated in dense and occluded environments such as urban environments. In this paper, we propose a multi-sensors placement solution for optimized coverage in dense urban environments. Our main contribution is based on two main efforts: 1. Defining conditions necessary for visibility, taking into account detection and false alarm rate probabilities, representing the sensor's stochastic character as part of our visibility analysis. 2. Unique concept facing partially visible objects, such as trees, in an urban scene, extending our previous work and proposing fast and exact 3D visible volumes analysis in urban scenes based on an analytic solution. We consider several 3D models for 3D visibility analysis and present an optimized solution using genetic algorithm, suited to our problem's constraints. We demonstrate the results through simulations with a 3D neighborhood model, taking trees into account. We demonstrate formulation of the conditions necessary for visibility related to detection and false alarm rate probabilities.

Keywords- *Visibility; 3D; Urban environment; Spatial analysis; Genetic algorithm; Sensor coverage.*

I. INTRODUCTION AND RELATED WORK

Modern cities and urban environments are becoming denser more heavily populated and are still rapidly growing, including new infrastructures, markets, banks, transportation, etc.

At the same time, security needs are becoming more and more demanding in our present era, in the face of terror attacks, crimes, and the need for improving law enforcement capabilities, as part of the increasing global social demand for efficient and immediate homeland and personal security in modern cities.

In the last two decades, more and more cities and megacities have started using multi-camera networks in order to face this challenge, mounting cameras for security monitoring needs [1]; however, this is still not enough [30]. Due to the complexity of working with 3D and the dynamic constraints of urban terrain, sensors were placed in busy and populated viewpoints, to observe the occurrences at these major points of interest.

These current multi-sensors placement solutions ignore some key factors, such as: visibility analysis in 3D models, which also consist of unique objects such as trees; changing the visibility analysis aspect from visible or invisible states to semi-visible cases, such as trees, and above all optimization solutions which take these factors into account.

Multi-sensor placement in 3D urban environments is not a simple task. The optimization problem of the optimal configuration of multi sensors for maximal coverage is a well-known Non-deterministic Polynomial-time hard (NP-hard) one [5], even without considering the added complexity of urban environments.

An extensive theoretical research effort has been made over the last four decades, addressing a much simpler problem in 2D known as the art gallery problem, with unrealistic assumptions such as unlimited visibility for each agent, while the 3D problem has not received special attention [8][28][35].

The coupling between sensors' performances and their environment's constraints is, in general, a complex optimization problem. In this paper, we study the multi-sensors placement optimization problem in 3D urban environments for optimized coverage based on genetic algorithms using novel visibility analysis.

Our optimization solution for this problem relates to maximal coverage from a number of viewpoints, where each 3D position (x, y, z coordinates) of the viewpoint is set as part of the optimized solution. The search space contains local minima and is highly non-linear. The Genetic Algorithms are global search methods, which are well-suited for such tasks. The optimization process is based on randomly generating an initial population of possible solutions (called chromosomes) and, by improving these solutions over a series of generations, it is able to achieve an optimal solution [36].

Multi-sensor placements are scene- and application-dependent, and for this reason generic rules are not very efficient at meeting these challenges. Our approach is based on a flexible and efficient analysis that can handle this complexity.

The total number of sensors is a crucial parameter, due to the real-time outcome data that should be monitored and tracked, where too many sensors are not an efficient solution.

We address the sensor numbers that should be set as a tradeoff of coverage area and logical data sources that can be monitored and tracked.

As part of our high-dimension optimization problem, we present several 3D models, such as B-ref, sweeping and wireframe models, Polyhedral Terrain Models (PTM) and Constructive Solid Geometry (CSG) for an efficient 3D visibility analysis method, integrating trees as part of our fast and efficient visibility computation, thus extending our previous work [25] to 3D visible volumes.

Accurate visibility computation in 3D environments is a very complicated process demanding a high computational effort, which cannot be easily carried out in a very short time using traditional well-known visibility methods [41]. The exact visibility methods are highly complex, and cannot be used for fast applications due to their long computation time. As mentioned above, previous research in visibility computation has been devoted to open environments using Digital Elevation Model (DEM) models, representing raster data in 2.5D (Polyhedral model), which do not address, or suggest solutions for, densely built-up areas.

One of the most efficient methods for DEM visibility computation is based on shadow-casting routine. The routine casts shadowed volumes in the DEM, like a light bubble [42]. Other methods related to urban design environment and open space impact treat abstract visibility analysis in urban environments using DEM, focusing on local areas and approximate openness [20]. Extensive research treated Digital Terrain Models (DTM) in open terrains, mainly Triangulated Irregular Network (TIN) and Regular Square Grid (RSG) structures. Visibility analysis on terrain was classified into point, line and region visibility, and several algorithms were introduced based on horizon computation describing visibility boundaries [11][12].

A vast number of algorithms have been suggested for speeding up the process and reducing computation time [38]. Franklin [21] evaluates and approximates visibility for each cell in a DEM model based on greedy algorithms. An application for siting multiple observers on terrain for optimal visibility cover was introduced in [23]. Wang et al. [52] introduced a Grid-based DEM method using viewshed horizon, saving computation time based on relations between surfaces and Line Of Sight (LOS), using a similar concept of Dead-Zones visibility [4]. Later on, an extended method for viewshed computation was presented, using reference planes rather than sightlines [53].

Most of these published papers have focused on approximate visibility computation, enabling fast results using interpolations of visibility values between points, calculating point visibility with the Line of Sight (LOS) method [13]. Other fast algorithms are based on the conservative Potentially Visible Set (PVS) [16]. These methods are not always completely accurate, as they may render hidden objects' parts as visible due to various simplifications and heuristics.

Only a few works have treated visibility analysis in urban environments. A mathematical model of an urban scene, calculating probabilistic visibility for a given object from a specific viewcell in the scene, has been presented by [37]. This is a very interesting concept, which extends the traditional deterministic visibility concept. Nevertheless, the buildings are modeled as cylinders, and the main challenges of spatial analysis and model building were not tackled. Other methods have been developed, subject to computer graphics and fields of vision, dealing with exact visibility in 3D scenes, without considering environmental constraints. Concerning this issue, Plantinga and Dyer [41] used the aspect graph – a graph with all the different views of an object. Shadow boundaries computation is a very popular method, studied by [14][47][48]. All of these works are not applicable to a large scene, due to computational complexity.

As mentioned, online visibility analysis is a very complicated task. Recently, off-line visibility analysis, based on preprocessing, was introduced. Cohen-Or et al. [4] used a ray-shooting sample to identify occluded parts. Schaufler et al. [44] use blocker extensions to handle occlusion.

Since visibility analysis in 3D urban environments is a very complicated task, it is therefore our main optimization function, known as Fitness. We introduce an extended visibility aspect for the common method of Boolean visibility values, "1" for objects seen and "0" for objects unseen from a specific viewpoint, and treat trees as semi-visibility values (such as partially seen, "0.5" value), thereby including in our analysis the real environmental phenomena, which are commonly omitted.

We extend our previous work and propose fast and exact 3D visible volumes analysis in urban scenes based on an analytic solution, integrating trees into our 3D model, and it is demonstrated with a real urban scene model from Neve-Sha'an neighborhood (within the city of Haifa).

In the following sections, we first introduce an overview of 3D models and our demands from these models. In the next section, we extended the 3D visible volumes analysis, which for the first time, takes trees into account. Later on, we present the simulation using the Neve-Sha'an neighborhood (within the city of Haifa) 3D model. We present our genetic algorithm optimization stages and simulation based on our 3D visible volumes analysis, taking trees into account. Eventually, we extend our current visibility aspect and include conditions necessary for visibility based on the sensor's stochastic character and present the effect of these limitations on our visibility analysis.

II. 3D MODELS FOR VISIBILITY ANALYSIS – OVERVIEW

In this section, we present a comprehensive overview of 3D models for urban scenes, from visibility analysis aspects. We divide the different models into polyhedral, parametric classes, which are available today using existing data sets, and examine the advantages and disadvantages of each. We

focus on visibility computation capabilities using these models.

A. Polyhedral models

Wireframe - In this model, 3D objects are represented as a set of vertices and lines, but not as faces. The model's assumption is that buildings consist of straight lines and that very dense scenes can be modeled. However, building types are very limited and, above all, the model is missing topological relations. Therefore, wireframe models are rarely used for visibility analysis applications.

B-rep - Boundary models offer a very flexible tool for modeling manmade objects. They are based on a surface-oriented view of solid objects: an object is considered as completely represented by its bounding faces. In order to represent the object correctly, boundary models consist of edges and vertices, as well as the topological relations of all features. The faces, edges, and vertices are the (labeled) nodes of a graph, and the direct neighborhood relations are described by a graph of edges, as shown in Figure 1.

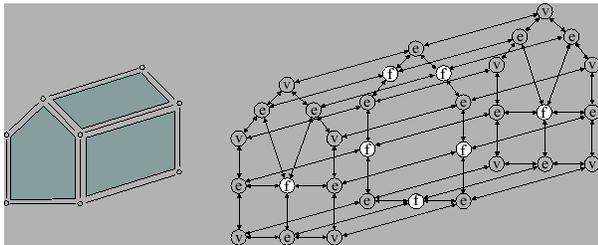


Figure 1. Boundary model of a solid object: a graph with nodes of type f (faces), e (edges) and v (vertices) and their topological relations (source: [34])

Boundary models are well-suited for visualization tasks because they readily include all data required for that purpose, which is why they are very often used for 3D solid modeling systems. On the other hand, simple operators demand a very complex computation effort, which is sometimes critical for efficient visibility analysis, with limited representation capabilities. Therefore, B-ref are very common for visualization but not for efficient visibility analysis.

B. Parametric models

Sweep methods: Sweep-representations of a 3D object are created by moving a planar (2D) shape, which is usually defined as a closed polygon, according to a pre-defined rule [43][46]. Depending on the rule by which the 2D shape is moved, two types of sweep representations can be distinguished, as seen in Figure 2:

Translational sweep: The shape is translated along a pre-defined translational vector.

Rotational sweep: The shape is rotated around a pre-defined rotational axis.

The concept of translational sweeps can be extended by sweeping two shapes along each other [34]. Sweep

representations are widely-used in computer vision, using symmetry for rendering techniques. However, topological relations and Boolean set operations between objects used in visibility methods such as *union*, *intersection* and *difference* are not supported. Moreover, the generation of arbitrary objects becomes rather difficult using this technique [46].

Constructive Solid Geometry (CSG): It is the aim of CSG to provide solid 3D primitives describing a set of parameters that reflect the object's dimensions. CSG primitives are simple objects such as cubes, boxes, tetrahedrons or quadratic pyramids. The CSG method can be easily adapted by using Boolean set operations (union, intersection and difference) in order to represent more complex objects consisting of more than one primitive, as shown in Figure 3. Therefore, CSG is the most useful and convenient method for visibility analysis, since the generation history of the solid itself, corresponding to the CSG tree upper node, is stored in the tree, as can be seen in Figure 4. As Boolean set operations are an integral part of a CSG tree, these operations are closed for CSG trees, e.g., the union of two CSG trees will again be a valid CSG tree [34].

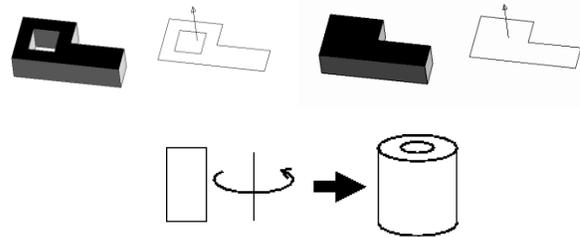


Figure 2. Sweeping a planar rectangular shape. (Top) a translational sweep creates a vertical prism. (Bottom) a rotational sweep creates a cylindrical object (source: [34])

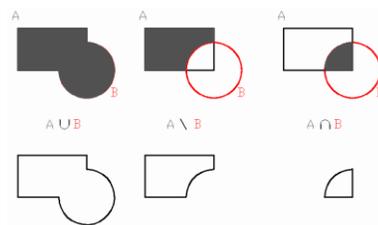


Figure 3. Boolean set operations (union, difference and intersection) (source: [34])

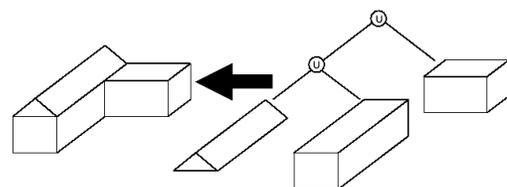


Figure 4. CSG - The CSG model (left) is represented by the CSG tree (right) consisting of three primitives connected by a Boolean union operation, (source: [34])

C. Discussion

Visibility analyses commonly use tree presentations, which allow fast Boolean operations and modeling many types of objects, where computational effort is a major issue. The existing models possess rules which eliminate them from representing the whole building's structure envisioned by the designing architect. Table I summarizes the different capabilities of each model for our demands, where CSG seems to be the most relevant model for visibility computation.

TABLE I. COMPARISONS OF 3D MODELS FOR VISIBILITY COMPUTATION

Model	Presentation friability	Fast Visibility Computation	Presentation Accuracy
Wireframe	Limited	Limited	Low
B-rep	Flexible	Limited	High
Sweep	Limited	Limited	Medium
CSG	Constraint free	Flexible	High

III. ANALYTIC 3D VISIBLE VOLUMES ANALYSIS

In this section, we present fast 3D visible volumes analysis in urban environments, based on an analytic solution that plays a major role in our proposed method of estimating the number of clusters. We briefly present our analysis presented in [27], extending our previous work [25] for surfaces' visibility analysis, and present an efficient solution for visible volumes analysis in 3D.

We analyze each building, computing visible surfaces and defining visible pyramids using analytic computation for visibility boundaries [25]. For each object we define Visible Boundary Points and (VBP) and Visible Pyramid (VP).

A simple case demonstrating analytic solution from a visibility point to a building can be seen in Figure 5(a). The visibility point is marked in black, the visible parts colored in red, and the invisible parts colored in blue where VBP are marked with yellow circles.

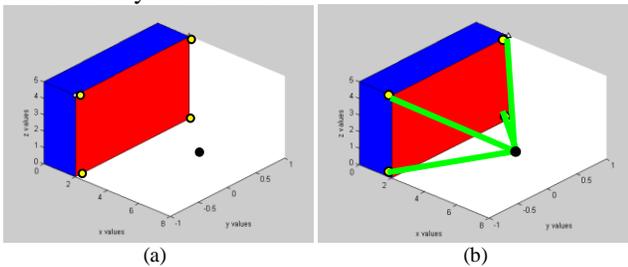


Figure 5. (a) Visibility Volume Computed with the Analytic Solution. (b) Visible Pyramid from a Viewpoint (marked as a Black Dot) to VBP of a Specific Surface (source: [27]).

In this section, we briefly introduce our concept for visible volumes inside bounding volume by decreasing visible pyramids and projected pyramids to the bounding volume boundary. First, we define the relevant pyramids and volumes.

The Visible Pyramid (VP): we define $VP_i^{j=1..N_{surf}}(x_0, y_0, z_0)$ of object i as a 3D pyramid generated by connecting VBP of specific surface j to a viewpoint $V(x_0, y_0, z_0)$.

In the case of a box, the maximum number of N_{surf} for a single object is three. VP boundary, colored with green arrows, can be seen in Figure 5(b).

For each VP, we calculate Projected Visible Pyramid (PVP), projecting VBP to the boundaries of the bounding volume S .

Projected Visible Pyramid (PVP) - we define $PVP_i^{j=1..N_{surf}}(x_0, y_0, z_0)$ of object i as 3D projected points to the bounding volume S , VBP of specific surface j through viewpoint $V(x_0, y_0, z_0)$. PVP boundary, colored with purple arrows, can be seen in Figure 6.

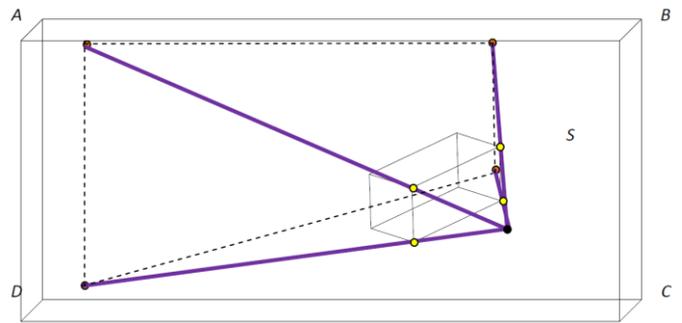


Figure 6. Invisible Projected Visible Pyramid Boundaries colored with purple arrows from a Viewpoint (marked as a Black Dot) to the boundary surface ABCD of Bounding Volume S (source: [27]).

The 3D Visible Volumes inside bounding volume S , VV_S , computed as the total bounding volume S , V_S , minus the Invisible Volumes IV_S . In a case of no overlap between buildings, IV_S is computed by decreasing the visible volume from the projected visible volume, $\sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j))$.

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} IV_{S_i^j} \tag{1}$$

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j))$$

By decreasing the invisible volumes from the total bounding volume, only the visible volumes are computed, as seen in Figure 7. Volumes of PVP and VP can be simply computed based on a simple pyramid volume geometric formula.

Invisible Hidden Volume (IHV) - We define Invisible Hidden Volume (IHV), as the Invisible Surface (IS) between visible pyramids projected to bounding box S .

The PVP of the object close to the viewpoint is marked in black, colored with pink circles denoted as boundary set points $\{B_{11}, \dots, B_{18}\}$ and the far object's PVP is colored with orange circles, denoted as boundary set points $\{C_{11}, \dots, C_{18}\}$. It can be seen that IHV is included in each of these invisible

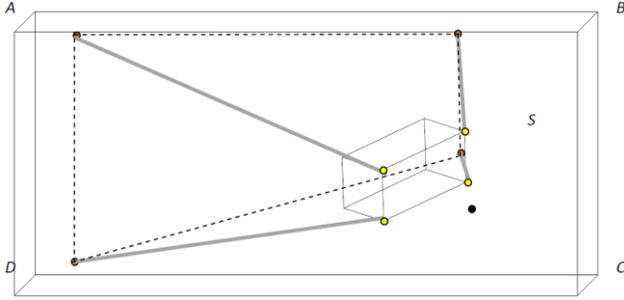


Figure 7. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ Colored in Gray Arrows. Decreasing Projected Visible Pyramid boundary surface ABCD of Bounding Volume S from Visible Pyramid (source: [27]).

volumes, where $\{A_{11}, \dots, A_{18}\} \in \{B_{11}, \dots, B_{18}\}$ and $\{A_{11}, \dots, A_{18}\} \in \{C_{11}, \dots, C_{18}\}$, as can be seen in Figure 8.

Therefore, we add IHV between each overlapping pair of objects to the total visible volume. In the case of overlapping between objects' visible pyramids, 3D visible volume is formulated as:

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) \quad (2)$$

The same analysis holds true for multiple overlapping objects, adding the IHV between each two consecutive objects.

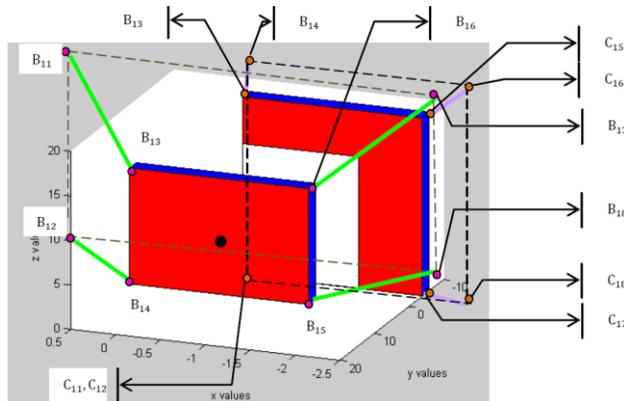


Figure 8. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ colored in purple and green arrows for each building. PVP of the object close to viewpoint colored in black, the far object PVP colored with orange circle (source: [27]).

Extended formulation for two buildings with or without overlap can be seen in [27].

A. Partial Visibility Concept - Trees

In this research, we analyze trees as constant objects in the scene, and formulate a partial visibility concept. In our previous work, we tested trees as dynamic objects and their effect on visibility analysis [26]. Still, the analysis focused on trees' branches over time, setting visible and invisible values for each state, taking into account probabilistic modeling in time.

We model trees as two boxes [40], as seen in Figure 9. The lower box, bounded between $[0, h_1]$ models the tree's trunk, leads to invisible volume and is analyzed as presented previously for a box modeling building's structures. On the other hand, the upper box bounded between $[h_1, h_2]$ is defined as partially visible, since a tree's leaves and the wind's effect are hard to predict and continuously change over time. Due to these inaccuracies, we set the projected surfaces and the Projected Visible Pyramid of this box as half visible volume.

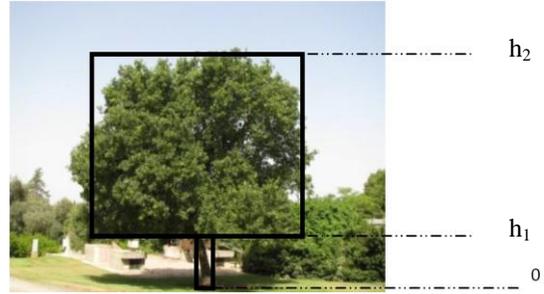


Figure 9. Modeling a Tree Using Two Bounding Boxes.

According to that, a tree's effect on our visibility analysis is divided into regular boxes included in the total number of objects, N_{obj} (identical to the building case), and the upper boxes modeling the tree's leaves, denoted as N_{trees} . The total 3D visible volumes can be formulated as:

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) - \sum_{i=1}^{N_{trees}} \sum_{j=1}^{N_{surf}} \frac{1}{2} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) \quad (3)$$

B. Simulations

In this section, we demonstrate our 3D visible volumes analysis in urban scenes integrated with trees, presented in the previous section. We have implemented the presented algorithm and tested some urban environments on a 1.8GHz Intel Core CPU with Matlab. Neve-Sha'an Street in the city of Haifa was chosen as a case study, presented in Figure 10.

We modeled the urban environment into structures using AutoCAD model, as seen in Figure 11, with bounding box S. By using the Matlab®MathWorks software we automated the transformation of data from AutoCAD structure to our model's internal data structure.

Our simulations focused on two cases: (1) small-scale housing in dense environments; (2) Multi-story buildings in an open area. These two different cases do not take the same objects into account. The first viewpoint is marked with black dot and the second one marked in purple, as seen in Figure 12. Since trees are not a part of our urban scene

model, trees are simulated based on similar urban terrain in Neve-Sha'an. We simulated fifty trees' locations using standard Gauss normal distribution, where the trees' parameters h_1, h_2 are defined randomly $h_1 \in (0.3, 0.9), h_2 \in (1.5, 3)$, as seen in Figure 12.



Figure 10. Views of Neve-Sha'an Street, Haifa, Israel from Google Maps source: [20]

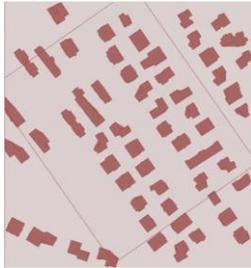


Figure 11. AutoCAD model of Neve-Sha'an Street, Haifa, Israel.

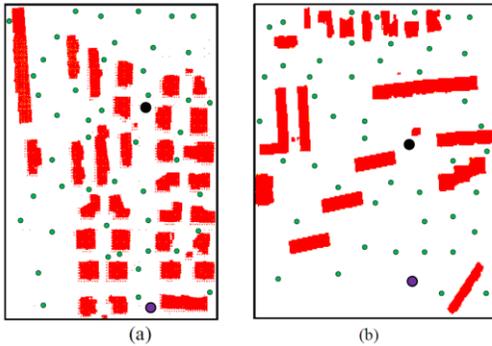


Figure 12. Tested Scenes with Trees marked with green points, Viewpoint 1 Colored in Black, Viewpoint 2 Colored in Purple : (a) Small-scale housing in dense environments; (b) Multi-story buildings in an open area.

We set two different viewpoints, and calculated the visible volumes based on our analysis presented in the previous sub-section. Visible volumes with time computation for different cases of bounding boxes' test scenes are presented in Table II and Table III.

One can notice that the visible volumes become smaller in the dense environments described in Table II, as we enlarge the bounding box. Since we take into account more buildings and trees, less volumes are visible and the total visible volumes from the same viewpoint are smaller. Pseudo-code of our visible volumes analysis can be seen in Section II.C.

TABLE II. VISIBLE VOLUMES AND COMPUTATION TIME FOR SMALL-SCALE HOUSING CASE

Bounding Box	Viewpoint	Visible Volumes [$10^5 \cdot m^3$]	Computation Time [sec]
[100 m * 100 m * 100 m]	Viewpoint 1	321.7	19.6
	Viewpoint 2	486.8	
[200 m * 200 m * 200 m]	Viewpoint 1	547.4	20.8
	Viewpoint 2	584.2	

TABLE III. VISIBLE VOLUMES FOR SMALL MULTI-STORY BUILDINGS CASE

Bounding Box [100 m * 100 m * 100 m]	Visible Volumes [$10^5 \cdot m^3$]	Computation Time [sec]
Viewpoint 1	3453	22.9
Viewpoint 2	3528	

C. 3D Visible Volumes - Pseudo Code

```

Given viewpoint  $V(x_0, y_0, z_0)$ 
1. Calculate bounding volume  $V_S$ 
2. For  $i=1:1:N_{obj}$  building models
    2.1. Calculate Azimuth  $\theta_i$  and Distance  $D_i$  from viewpoint to object
    2.2. Set and Sort Buildings Azimuth Array  $\theta[i]$ 
    2.3. IF Azimuth Objects  $(i, 1..i-1)$  Intersect
        2.3.1. Sort Intersected Objects  $j=1:1:N_{intersect}$  by Distance
        2.3.2. Compute VBP for each intersected building,  $VBP_{j=1..N_{intersect}}^{1..N_{bound}}$ 
        2.3.3. Generate VP for each intersected building,  $VP_{j=1..N_{intersect}}^{1..N_{surf}}$ 
        2.3.4. Set  $PVP_i^j$  and  $IHW_i^j$  volumes for objects,  $N_{obj}$ 
        2.3.5. Set  $PVP_i^j$  and  $IHW_i^j$  volumes for Trees,  $N_{Trees}$ 
    Else
        2.3.6. Compute VBP for each object,  $VBP_{j=1..N_{intersect}}^{1..N_{bound}}$ 
        2.3.7. Generate VP for each building,  $VP_{j=1..N_{intersect}}^{1..N_{surf}}$ 
        2.3.8. Set  $PVP_i^j$  volumes for objects,  $N_{obj}$ 
        2.3.9. Set  $PVP_i^j$  volumes for Trees,  $N_{Trees}$ 
    End
2.4. Calculate Visible Volumes  $VV_S$ 
End

```

D. Complexity Analysis

We analyze our algorithm complexity based on the pseudo code presented in the previous section, where n represents the number of buildings and trees. In the worst case, n objects hide each other. Visibility complexity consists of generating VBP and VP for n objects, $nO(1)$ complexity. Projection and intersection are also $nO(1)$ complexity. The complexity of our algorithm, without considering data structure managing for urban environments, is $nO(n)$.

IV. OPTIMIZED COVERAGE USING GENETIC ALGORITHMS

The Genetic Algorithm (GA) presented by Holland [31] is one of the most common algorithms from the evolutionary

algorithms class used for complex optimization problems in different fields, such as: pharmaceutical design [33], financial forecasting [50], tracking and coverage [18][39][45], and bridge design [24]. These kinds of algorithms, inspired by natural selection and genetics, are sometimes criticized for their lack of theoretical background due to the fact that in some cases the outcome is unpredictable or difficult to verify.

The main idea behind GA is based on repeated evaluation of individuals (which are part of a candidate solution) using an objective function over a series of generations. These series are improved over generations in order to achieve an optimal solution. In the next paragraphs, we present the genetic algorithms' main stages, adapted to our specific problem.

The major stages in the GA process (evaluation, selection, and reproduction) are repeated either for a fixed number of generations, or until no further improvement is noted. The common range is about 50-200 generations, where fitness function values improve monotonically [31]. A block diagram of GA is depicted in Figure 13.

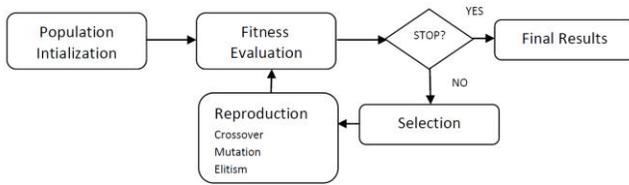


Figure 13. GA Block Diagram, source: [31].

Population Initialization: The initialization stage creates the first generation of candidate solutions, also called chromosomes. A population of candidate solutions is generated by a random possible solution from the solution space. The number of individuals in the population is dependent on the size of the problem and also on computational capabilities and limitations. In our case, it is defined as 500 chromosomes, due to the fact that 3D visible volumes must be computed for each candidate.

For our case, the initialized population of viewpoints configuration is set randomly, and would probably be a poor solution due to its random nature, as can be estimated. The chromosome is a 3xN-dimensional vector for N sensor's locations, i.e., viewpoints, where position and translation is a 3-dimensional (x,y,z) vector for each viewpoint location, as seen in Figure 14. The population is depicted in Figure 15.

Evaluation: The key factor of genetic algorithm relates to individual evaluation, which is based on a score for each chromosome, known as Fitness function. This stage is the most time-consuming in our optimization, since we evaluate all individuals in each generation. It should be noticed that each chromosome score leads to 3D visible volume computation N times. As a tradeoff between the covered area and computational effort, we set N to eight. In the worst case, one generation evaluation demands visibility analysis for four thousand different viewpoints. In such a case, one

can easily understand the major drawback of the GA method in relation to computational effort. Nevertheless, parallel computation has made a significant breakthrough over the last two decades; GA and other optimization methods based on independent evaluation of each chromosome can nearly be computed in linear time.

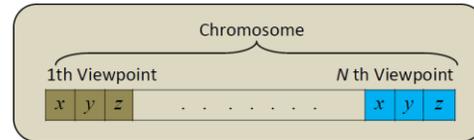


Figure 14. An individual in the GA search is also called "Chromosome". In our case it represents one possible sensor's location for N viewpoints computing 3D visible volumes analysis with trees.

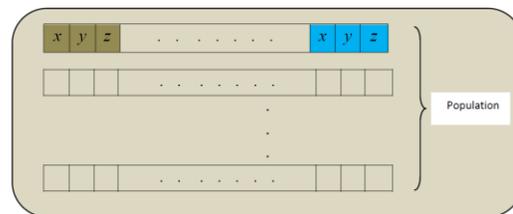


Figure 15. Population of GA search with N chromosomes.

Fitness Function: The fitness function evaluates each chromosome using optimization function, finding a global minimum value, which allows us to compare chromosomes in relation to each other.

In our case, we evaluate each chromosome's quality using 3D visible volumes normalized to the bounding box S around a viewpoint:

$$f(i) = \frac{1}{S} \sum_{j=1}^N VV_s(x_j, y_j, z_j) \tag{4}$$

Selection: Once the population is sorted by fitness, chromosomes' population with greater values will have a better chance of being selected for the next reproduction stage. Over the last years, many selection operators have been proposed, such as the Stochastic Universal Sampling and Tournament Selection. We used the most common Tournament, where k individuals are chosen randomly, and the best performance from this group is selected. The selection operator is repeated until a sufficient number of parents are chosen to form a child generation.

Reproduction: In this stage, the parent individuals chosen in the previous step are combined to create the next generation. Many types of reproduction have been presented over the years, such as crossover, mutation and elitism.

Crossover takes parts from two parents and splices them to form two offspring, as seen in Figure 16(a). Mutation modifies the parameters of a randomly selected chromosome from within a single parent, as seen in Figure 16(b). Elitism takes the fittest parents from the previous generation and replicates them into the new generation. Finally, individuals not selected as parents are replaced with new, random

offspring. Further analysis and operators can be found in [29][36]. The major steps of these operators can be seen in Figure 16.

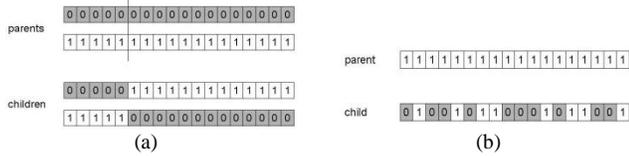


Figure 16. Reproduction operators of GA (a) Crossover (b) Mutation source: [17].

A. Simulations

In this section, we report on simulation runs with our 3D visible volumes analysis in urban scenes integrated with trees, using genetic algorithms. The genetic algorithms were tested on a 1.8GHz Intel Core CPU with Matlab. We used Fallvile Island Sketchup Google Model [19] for simulating a dense urban scene with trees, as seen in Figure 17.

The stages of Crossover and Elitism operators are described as follows, with a probability of $p_c = 0.9$ (otherwise parents are copied without change):

1. Choose a random point on the two parents.
 2. Split parents at this crossover point.
 3. Create next generation chromosomes by exchanging tails.
- Where the Mutation operator modifies each gene independently with a probability of $p_m = 0.1$.

In order to process the huge amount of data, we bounded a specific region, which includes trees and buildings, as seen in Figure 18. We imported the chosen region to Matlab and modeled the objects by boxes, neglecting roofs' profiles. Time computation for one generation was one hour long on average. As we could expect, the evaluation stage took up 94% of the total simulation time. We set the bounding box S as [500 m* 200 m* 50 m]. Population initialization included 500 chromosomes, each of which is a 24-dimensional vector consisting of position and translation, where all of them were generated randomly.

Based on the Fitness function described previously and the different GA stages and 3D visible volumes analysis, the location of eight viewpoints for sensor placement was optimized. Viewpoints must be bounded in S and should not penetrate buildings and trees. Stop criteria was set to 50 generations and Fitness function gradient.

Optimal coverage of viewpoints and visible volumes during ten runnings' simulations is seen in Figure 19, bounded in polygons marked with arrows. During these ten runnings simulations, we initialized the population randomly at different areas inside bounding box S.

These interesting results show that trees' effect inside a dense urban environment was minor, and trees around the buildings in open spaces set the viewpoint's location. As seen in Figure 19, polygon A and polygon B are both outside the areas blocked by buildings. But they are still located near trees, which affect the visible volumes, and we can predict that the same affect will occur in our real world. On the other

hand, polygon C, which is closer to the area blocked by buildings, takes into account the trees in this region, but the major factor are still the buildings.



Figure 17. Fallvile Island Sketchup Google Model Simulating Dense Urban Scene with Trees, [19]: (a) Topview; (b) Isometric view.



Figure 18. Bounded Area inside Bounding Box S marked in Black, inside Fallvile Island Sketchup Google Model.



Figure 19. Bounded Polygons of Optimized Cover Viewpoints Using GA marked with Arrows.

V. VISIBILITY ANALYSIS CONSIDERING SENSOR'S STOCHASTIC CHARACTER

In this section, we extend our visibility model by exploring and including sensors' sensing capabilities and physical constraints. Our visibility analysis is based on the fact that sensors are located at specific visibility points. Sensors are commonly treated as deterministic detectors, where a target can only be detected or undetected. These simplistic sensing models are based on the disc model [6][49].

We study sensors' visibility-based placement effected by taking into account the stochastic character of target detection. We present a single sensor model, including noisy measurement, and define the necessary condition for visibility analysis with false alarm and detection probabilities for each visibility point's candidate.

A. Single Visibility Sensing Model

Most of the physical signals are based on energy vs. distance from single source model. Different kind of sensors such as: radars, lasers, acoustics, etc., are based on this signal character. Like other signal models, presented in the literature [15][32][51] we use signal decay model as follows:

$$L(d) = \begin{cases} \frac{L_0}{(\frac{d}{d_0})^k}, & \text{if } d > d_0 \\ L_0, & \text{if } d \leq d_0 \end{cases} \quad (5)$$

where L_0 is the original energy emitted by the target, k is the decaying factor (typical values from 2 to 5), and d_0 is a constant determined by the size of the target and the sensor.

We model the sensor's noise N_i located at visibility point V_i , using zero-mean normal distribution, $N_i \sim N(0, \sigma^2)$. Sensor signal energy including noise effect, S_i , can be formulated as:

$$S_i = L(d_i) + N_i^2 \quad (6)$$

In practice, S_i parameters are set by empiric datasets.

B. Visibility Using Sensors Network

Nowadays, detection systems use more and more data fusion methods [9][10]. In order to use multi sensors benefits, fusion and local decision-making using several sensors' data is a very common capability. As with other distributed data fusion methods, we assume that each sensor sends the energy measurement to a Local Decision Making Module (LDMM). Similar to other well known fusion methods [51], the LDMM integrates and compares the average sensors' measurements n against *detection threshold* τ .

Detection probability, denoted by P_D , is the probability that a target is correctly detected. Supposing that n sensors

take part in the data fusion applied in the LDMM, detection probability is given by:

$$P_D = P\left(\frac{1}{n} \sum_{i=1}^n (L(d_i) + N_i^2) > \tau\right)$$

$$P_D = 1 - P\left(\sum_{i=1}^n \left(\frac{N_i}{\sigma}\right)^2 \leq \frac{n\tau - \sum_{i=1}^n L(d_i)}{\sigma^2}\right) \quad (7)$$

$$P_D = 1 - X_n\left(\frac{n\tau - \sum_{i=1}^n L(d_i)}{\sigma^2}\right)$$

Where $N_i/\sigma \sim N(0,1)$ and X_n denote the distribution function. In the same way, false alarm rate probability is the probability of making a positive detection decision when no target is present. False alarm rate probability, denoted by P_F , is given by:

$$P_F = P\left(\frac{1}{n} \sum_{i=1}^n N_i^2 > \tau\right) = 1 - P\left(\sum_{i=1}^n \left(\frac{N_i}{\sigma}\right)^2 \leq \frac{n\tau}{\sigma^2}\right) \quad (8)$$

$$P_F = 1 - X_n\left(\frac{n\tau}{\sigma^2}\right)$$

Conditions Necessary for Visibility: Given two real numbers, $a \in (0,1)$ and $b \in (0,1)$. Visibility Point $V_i(x, y, z)$ can be defined as visible point **if and only if** $P_F(V_i) \leq a$ and $P_D(V_i) \geq b$.

The conditions necessary for visibility plays a major role in the GA process. In order to include stochastic sensor character as part of our visibility analysis and sensor placement, we suggest an updated GA search block diagram. As described above, the population stage creates the first generation of candidate solutions, also called chromosomes. These chromosomes should be tested and pass the necessary condition, as can be seen in Figure 20. If a specific chromosome fails, other chromosomes are generated randomly as part of the population initialization stage.

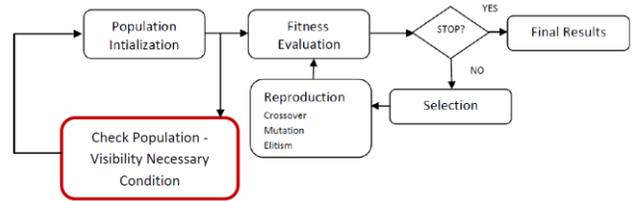


Figure 20. GA Block Diagram Including Conditions Necessary for Visibility.

C. Simulations

In this section, we report on simulation runs including conditions necessary for visibility as part of our genetic algorithms search, according to the block diagram presented in Figure 20. In the same manner, similar to the simulation environment presented in Section IV.A, we used 1.8GHz Intel Core CPU with Matlab using Fallville Island Sketchup Google Model [14], simulating a dense urban scene with trees, as seen in Figure 17.

In order to compare our current running results with the case of not using conditions necessary for visibility, which are detailed in Section IV.A., we used, in our case, the exact same running parameters of the genetic algorithm search in the different stages (population, evaluation, selection, reproduction, fitness function and stopping criteria).

Sensing model of detection and false alarm rate probabilities set the sensor's detection performances, and have a great influence on which objects are included in the bounding box. Our parameters of the visibility sensing model were set as follows: $L_0=1$, $d_0=35$, $k=2$, $\sigma^2 = 0.1$.

As expected, simulation results applying the conditions necessary for visibility generated results similar to the case of not using this condition, as the detection probability is set to higher values with lower values of false alarm rate probability. On the same way, viewpoints' locations were bounded in the A, B, C polygons described in Figure 19. Results can be seen in Table IV.

TABLE IV. VIEWPOINTS LOCATIONS DIFFERENCES USING CONDITIONS NECESSARY FOR VISIBILITY USING GA SEARCH

Detection Probability P_D	False Alarm Rate Probability P_F	Detected Objects in Bounding Box [Percents]	Viewpoints Located Outside A, B, C Polygons
0.9	0.01	87%	3
0.95	0.005	94%	2
0.98	0.002	96%	1
0.99	0.001	97%	1

VI. CONCLUSIONS

In this paper, we presented an optimized solution for the problem of computing maximal coverage from a number of viewpoints, using genetic algorithms method. In addition, we propose conditions necessary for visibility based on sensors' model analysis, taking into account stochastic character. As far as we know, for the first time we integrated trees as partially visible objects participating in a 3D visible volumes analytic analysis and conditions necessary for visibility with sensors' noises effects. As part of our research we tested several 3D models of 3D urban environments from the visibility viewpoint, choosing the best model from the computational effort and the analytic formulation aspects.

We tested our 3D visible volumes method on real a 3D model from an urban street in the city of Haifa, with time computation and visible volumes parameters.

In the second part of the paper, we introduced a genetic algorithm formulation to calculate an optimized solution for the visibility problem. We used several reproduction operators, which made our optimization robust. We tested our algorithm on the Fallville Island Sketchup Google Model combined with trees, and analyzed the viewpoint's polygons results, and also compared using versus not using the conditions necessary for visibility.

Our future work is related to validation between our simulated solution and projected volumes from sensors mounted in these viewpoints for optimal coverage.

VII. REFERENCES

- [1] O. Gal and Y. Doytsher, "Sensors Placement in Urban Environments Using Genetic Algorithms," The Senventh International Conference on Advanced Geographic Information Systems, Applications, and Services, pp. 82-87, 2015.
- [2] N. Abu-Akel, "Automatic Building Extraction Using LiDAR Data," PhD Dissertation, Technion, Israel, 2010.
- [3] D. Cohen-Or, G. Fibich, D. Halperin, and E. Zadicario, "Conservative Visibility and Strong Occlusion for Viewspace Partitioning of Densely Occluded Scenes," In EUROGRAPHICS'98.
- [4] D. Cohen-Or and A. Shaked, "Visibility and Dead- Zones in Digital Terrain Maps," Eurographics, vol. 14, no. 3, pp.171-180, 1995.
- [5] R. Cole and M. Sharir, "Visibility Problems for Polyhedral Terrain," Journal of Symbolic Computation, vol. 7, pp.11-30, 1989.
- [6] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," IEEE Trans. Comput, vol. 51, no. 12, 2002.
- [7] Y. Chrysanthou, "Shadow Computation for 3D Interactive and Animation," Ph.D. Dissertation, Department of Computer Science, College University of London, UK, 1996.
- [8] R. Church and C. ReVelle, "The Maximal Covering Location Problem," Papers of the Regional Science Association, vol. 32, pp.101-118, 1974.
- [9] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K.K. Saluja, "Sensor deployment strategy for target detection," In WSNA, 2002.
- [10] T. Clouqueur, K.K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," IEEE Trans. Comput, vol. 53, no. 3, 2004.
- [11] L. De Floriani and P. Magillo, "Visibility Algorithms on Triangulated Terrain Models," International Journal of Geographic Information Systems, vol. 8, no. 1, pp.13-41, 1994.
- [12] L. De Floriani and P. Magillo, "Intervisibility on Terrains," In P.A. Longley, M.F. Goodchild, D.J. Maguire & D.W. Rhind (Eds.), Geographic Information Systems: Principles, Techniques, Management and Applications, pp. 543-556. John Wiley & Sons, 1999.
- [13] Y. Doytsher and B. Shmutter, "Digital Elevation Model of Dead Ground," Symposium on Mapping and Geographic Information Systems (ISPRS Commission IV), Athens, Georgia, USA, 1994.
- [14] G. Drettakis and E. Fiume, "A Fast Shadow Algorithm for Area Light Sources Using Backprojection," In Computer Graphics (Proceedings of SIGGRAPH '94), pp. 223-230, 1994.
- [15] M.F. Duarte and Y.H. Hu, "Vehicle classification in distributed sensor networks," Journal of Parallel and Distributed Computing, vol. 64, no. 7, 2004.
- [16] F. Durand, "3D Visibility: Analytical Study and Applications," PhD thesis, Universite Joseph Fourier, Grenoble, France, 1994.
- [17] A.E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing Genetic Algorithms," Lecture Notes, 1999.
- [18] U.M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," Computer Vision and Image Understanding, vol. 103, no. 3, pp. 156-169, 2006.
- [19] Fallville, (2010) <http://sketchup.google.com/3dwarehouse/details?mid=2265cc05839f0e5925ddf6e8265c857c&prevstart=0>

- [20] D. Fisher-Gewirtzman, A. Shashkov, and Y. Doytsher, "Voxel Based Volumetric Visibility Analysis of Urban Environments," *Survey Review*, DOI: 10.1179/1752270613Y.0000000059, 2013.
- [21] W.R. Franklin, "Siting Observers on Terrain," in D. Richardson and P. van Oosterom, eds, *Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling*, Springer-Verlag, pp. 109-120, 2002.
- [22] W.R. Franklin and C. Ray, "Higher isn't Necessarily Better: Visibility Algorithms and Experiments," In T. C. Waugh & R. G. Healey (Eds.), *Advances in GIS Research: Sixth International Symposium on Spatial Data Handling*, pp. 751-770. Taylor & Francis, Edinburgh, 1994.
- [23] W.R. Franklin and C. Vogt, "Multiple Observer Siting on Terrain with Intervisibility or Loos Data," in XXth Congress, International Society for Photogrammetry and Remote Sensing, Istanbul, 2004.
- [24] H. Furuta, K. Maeda, and E. Watanabe, "Application of Genetic Algorithm to Aesthetic Design of Bridge Structures," *Computer-Aided Civil and Infrastructure Engineering*, vol. 10, no. 6, pp.415-421, 1995.
- [25] O. Gal and Y. Doytsher, "Analyzing 3D Complex Urban Environments Using a Unified Visibility Algorithm," *International Journal On Advances in Software*, ISSN 1942-2628, vol. 5 no.3&4, pp:401-413, 2012.
- [26] O. Gal and Y. Doytsher, "Dynamic Objects Effect on Visibility Analysis in 3D Urban Environments," *Lecture Notes in Computer Science (LNCS)*, vol. 7820, pp.147-163, DOI 10.1007/978-3-642-37087-8_11, Springer, 2013.
- [27] O. Gal and Y. Doytsher, "Spatial Visibility Clustering Analysis In Urban Environments Based on Pedestrians' Mobility Datasets," *The Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services*, pp. 38-44, 2014.
- [28] H. González-Bañós and J.C. Latombe, "A randomized art-gallery algorithm for sensor placement," in *ACM Symposium on Computational Geometry*, pp. 232-240, 2001.
- [29] D.E. Goldberg and J.H. Holland, "Genetic Algorithms and Machine Learning," *Machine Learning*, vol. 3, pp.95-99, 1998.
- [30] Holden (2013), <http://slog.thestranger.com/slog/archives/2013/04/21/after-the-boston-bombings-do-american-cities-need-more-surveillance-cameras>
- [31] J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, 1975.
- [32] D. Li and Y.H. Hu, "Energy based collaborative source localization using acoustic micro-sensor array," *EUROSIP J. Applied Signal Processing*, vol. 4, 2003.
- [33] D. Maddalena and G. Snowden, "Applications of Genetic Algorithms to Drug Design," In *Expert Opinion on Therapeutic Patents*, pp. 247-254, 1997.
- [34] M. Mäntylä M, "An Introduction to Solid Modeling," Computer Science Press Rockville, Maryland, 1988.
- [35] M. Marengoni, B.A. Draper, A. Hanson, and R. Sitaraman, "A System to Place Observers on a Polyhedral Terrain in Polynomial Time," *Image and Vision Computing*, vol. 18, pp.773-780, 2000.
- [36] M. Mitchell, "An Introduction to Genetic Algorithms (Complex Adaptive Systems)," MIT Press, 1998.
- [37] B. Nadler, G. Fibich, S. Lev-Yehudi D., and Cohen-Or, "A Qualitative and Quantitative Visibility Analysis in Urban Scenes," *Computers & Graphics*, vol. 5, pp.655-666, 1999.
- [38] G. Nagy, "Terrain Visibility, Technical report," Computational Geometry Lab, ECSE Dept., Rensselaer Polytechnic Institute, 1994.
- [39] K.J. Obermeyer, "Path Planning for a UAV Performing Reconnaissance of Static Ground Targets in Terrain," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, PP.1-11, 2009.
- [40] K. Omasa, F. Hosoi, T.M. Uenishi, Y. Shimizu, and Y. Akiyama, "Three-Dimensional Modeling of an Urban Park and Trees by Combined Airborne and Portable On-Ground Scanning LIDAR Remote Sensing," *Environ Model Assess* vol. 13, pp.473-481, DOI 10.1007/s10666-007-9115-5, 2008.
- [41] H. Plantinga and R. Dyer, "Visibility, Occlusion, and Aspect Graph," *The International Journal of Computer Vision*, vol. 5, pp.137-160, 1990.
- [42] C. Ratti, "The Lineage of Line: Space Syntax Parameters from the Analysis of Urban DEMs," *Environment and Planning B: Planning and Design*, vol. 32, pp.547-566, 2005.
- [43] H. Samet, "Hierarchical Spatial Data Structures," *Symposium on Large Spatial Databases (SSD)*, pp. 193-212, 1989.
- [44] G. Schaufler, J. Dorsey, X. Decoret, and F.X. Sillion, "Conservative Volumetric Visibility with Occluder Fusion," In *Computer Graphics, Proceedings of SIGGRAPH*, pp. 229-238, 2000.
- [45] V. Shaferman and T. Shima, "Coevolution genetic algorithm for UAV distributed tracking in urban environments," in *ASME Conference on Engineering Systems Design and Analysis*, 2008.
- [46] A. Streilein, "Digitale Photogrammetrie und CAAD," Ph.D Thesis, Swiss Federal Institute of Technology (ETH) Zürich, Diss. ETH Nr. 12897, Published in *Mitteilungen Nr. 68 of the Institute of Geodesy and Photogrammetry*, 1999.
- [47] J. Stewart and S. Ghali, "Fast Computation of Shadow Boundaries Using Spatial Coherence and Backprojections," In *Computer Graphics, Proceedings of SIGGRAPH*, pp. 231-238, 1994.
- [48] S.J. Teller, "Computing the Antipenumbra of an Area Light Source," *Computer Graphics*, vol. 26, no.2, pp.139-148, 1992.
- [49] D. Tian and N.D. Georganas, "A coverage-preserved node scheduling scheme for large wireless sensor networks," In *WSNA*, 2002.
- [50] E.P.K. Tsang and J. Li, "Combining Ordinal Financial Predictions with Genetic Programming," In *Proceedings of the Second International Conference on Intelligent Data Engineering and Automated Learning*, pp. 532-537, 2000.
- [51] P. Varshney, "Distributed Detection and Data Fusion," Spinger-Verlag, 1996.
- [52] J. Wang, G.J. Robinson, and K. White, "A Fast Solution to Local Viewshed Computation Using Grid-based Digital Elevation Models," *Photogrammetric Engineering & Remote Sensing*, vol. 62, pp.1157-1164, 1996.
- [53] J. Wang, G.J. Robinson, and K. White K, "Generating Viewsheds without Using Sightlines," *Photogrammetric Engineering & Remote Sensing*, vol. 66, pp. 87-90, 2000.

Study the Throughput Outcome of Desktop Cloud Systems Using DesktopCloudSim Tool

Abdulelah Alwabel, Robert Walters, Gary Wills

School of Electronics and Computer Science

University of Southampton

Southampton, UK

e-mail: {aa1a10, rjw1, gbw}@ecs.soton.ac.uk

Abstract— Desktop Cloud computing is a new type of Cloud computing that aims to provide Cloud services at little or no cost. This ambition can be achieved by combining Cloud computing and Volunteer computing into Desktop Clouds, harnessing non-dedicated resources when idle. However, Desktop Cloud systems suffer from the issue of node failures. Node failure can happen without prior notification, which may affect the throughput outcome of these systems. This paper studies the impact of node failures using a simulation tool. Simulation tools are commonly used by academics and researchers to simulate Clouds in order to investigate various research issues and examine proposed solutions. CloudSim is a well-known and widely employed tool to simulate Cloud computing by both academia and industry. However, CloudSim lacks the ability to simulate failure events, which may occur to physical nodes in the infrastructure level of a Cloud system. In order to show the effectiveness of DesktopCloudSim, we evaluate the throughput of two types of Desktop Clouds: private and public Desktop Clouds that are built on top of faulty nodes based on empirical data sets. The data sets are analysed and studied in this paper to reflect the number of node failures in these two Cloud types. The evaluation process serves two purposes: the first is that it validate the working of the proposed tool. The second is to show that throughput of Desktop Cloud systems is affected badly by node failures.

Keywords-Cloud; CloudSim; DesktopCloudSim; Failure; Nodes; Throughput; VM Allocation.

I. INTRODUCTION

DesktopCloudSim [1] is proposed in our previous paper as an extension tool that can simulate node failures in Cloud system. Cloud computing has emerged with a promise to improve performance and reduce running costs. The services of Cloud computing are provided by Cloud service providers (CSPs). Traditionally, CSPs use a huge number of computing resources in the infrastructure level located in datacentres. Such resources are claimed to have a high level of reliability, which makes them resilient to failure events [2]. However, a new direction of Cloud has recently emerged with an aim to exploit normal Desktop computers, laptops, etc. to provide Cloud services [3]. This kind of Cloud can be called Desktop Clouds [4]. In contrast to the traditional way

of CSP, which uses a huge number of computing resources that are dedicated to be part of the Cloud. Throughout this paper, the term Traditional Cloud refers to this traditional way of Clouds.

The cost-effectiveness of Desktop Clouds is the key advantage over Traditional Clouds. Researchers in Desktop Clouds can use Cloud services at little cost, if not free. However, such feature suffers from an issue. The nodes of a Desktop Cloud are quite volatile and prone to failure without prior knowledge. This may affect the throughput of tasks and violate the service level agreement. The throughput is defined as the number of successful tasks submitted to be processed by virtual machines (VMs). Various VM allocation mechanisms can yield different variations of throughput level in the presence of node failures.

VM allocation mechanism is the process of allocation requested VMs by Cloud's users to physical machines (PMs) in the infrastructure level of a Cloud. The main goal of this paper is to study the impact of node failures on the outcome of Desktop Cloud systems. The contribution of this paper can be summarised into: (i) it proposes and describes the DesktopCloudSim as being an extension for CloudSim simulation toolkit; (ii) it investigates the impact of failure events on throughput and (iii) three VM mechanisms: FCFS, Greedy and RoundRobin mechanisms are evaluated in terms of throughput using DesktopCloudSim. The reminder of this paper is organised as follows: Section II discusses Desktop Cloud as being a new direction of Cloud computing. Section III proposes the simulation tool that extends CloudSim. The section starts by reviewing CloudSim to show the need to extend it. The section, then, reviews some VM allocation mechanisms. Next section demonstrates experiments conducted to evaluate the impact of node failures in a private Desktop Cloud based on empirical data of failures in NotreDame nodes. Another simulation of public Desktop Cloud is conducted using data of SETI@home nodes. The results are then analysed and discussed in Section V. Several related works are reviewed in Section VI. Finally, a conclusion and future work insights are given in the last section.

II. DESKTOP CLOUD

The success of Desktop Grids stimulates the idea of harnessing idle computer machines to build Desktop Clouds. Hence, the term Desktop comes from Desktop Grids because both of Desktop Clouds and Desktop Grids are based on Desktop PCs and laptops etc. Similarly, the term Cloud comes from Cloud as Desktop Cloud aims to provide services based on the Cloud business model. Several synonyms for Desktop Cloud have been used, such as Ad-hoc Cloud [5], Volunteer Cloud [3], Community Cloud [6] and Non-Dedicated Cloud [7]. The literature indicates that very little work has been undertaken in this direction.

Table I. Traditional Clouds vs. Desktop Clouds

Feature	Traditional Clouds	Desktop Clouds
Resources	Dedicated	Non-dedicated and volatile
Cost	Relatively high	Cheap
Location	Limited to a number of data centres	Distributed across the globe
Services	Reliable and available	Low availability and unreliable
Heterogeneity	Heterogeneous	Very heterogeneous

Desktop Clouds differ from Traditional Clouds in several things, as it is depicted in Table I. Firstly, the infrastructure of Desktop Cloud consists of resources that are non-dedicated, i.e., not made to be part of Cloud infrastructure. Desktop Cloud helps in saving energy since it utilises already-running undedicated resources, which would otherwise remain idle. Some studies show that the average percentage of local resources being idle within an organisation is about 80% [8]. It is shown that an idle machine can consume up to 70% of the total power consumed when it is fully utilised according to [9]. On the contrary, the infrastructure of Traditional Clouds is made of a large number of dedicated computing resources. Traditional Clouds have a negative impact on the environment since their data centres consume massive amounts of electricity for cooling these resources.

Secondly, resources of Desktop Clouds are quite scattered across the globe, whereas they are limited in Traditional Cloud to a number of locations in data centres. Furthermore, nodes in Desktop Cloud are highly volatile because nodes of Desktop Clouds can be down unexpectedly without prior notice. Node failures can occur for various reasons such as connectivity issues, machine crashing or simply the machine becomes busy with other work by its owner takes priority. High volatility in resources has negative impact on availability and performance [10]. Although, resources in both Traditional Cloud and Desktop Cloud are heterogeneous, they are even more heterogeneous and dispersed in Desktop Cloud. Traditional Clouds are centralised, which leads to the potential that there could be a single point of failure issue if a Cloud service provider goes out of the business. In contrast, Desktop Clouds manage and

offer services in a decentralised manner. Virtualisation plays a key role in both Desktop Clouds and Traditional Clouds.

Desktop Clouds can be confused with other distributed systems, specifically Desktop Grids. Both Desktop Clouds and Desktop Grids share the same concept that is exploiting computing resources when they become idle. The resources in both systems can be owned by an organisation or denoted by the public over the Internet. Both Desktop Grids and Desktop Clouds can use similar resources. Resources are volatile and prone to failure without prior knowledge. However, Desktop Grids differ from Desktop Clouds in the service and virtualisation layers. Services, in Desktop Clouds, are offered to clients in an elastic way. Elasticity means that users can require more computing resources in short term [11]. In contrast, the business model in Desktop Grids is based on a 'project oriented' basis, which means that every user is allocated a certain time to use a particular service [12]. In addition, Desktop Grids' users are expected to be familiar with details about the middleware used in order to be able to harness the offered services [13]. Specific software needs to be installed to computing machines in order to join a Desktop Grid. Clients in Desktop Clouds are expected to have little knowledge to enable them just use Cloud services under the principle ease of use. Desktop Grids do not employ virtualisation to isolate users from the actual machines while virtualisation is highly employed in Desktop Clouds to isolate clients from the actual physical machines.

III. DESKTOPCLOUDSIM

DesktopCloudSim is an extension tool proposed to simulate failure events happening in the infrastructure level based on CloudSim simulation tool. Therefore, this section starts by a brief discussion of CloudSim. The extension tool, DesktopCloudSim, is presented next. DesktopCloudSim is used to evaluate VM allocation mechanisms, thus the last subsection in this section discusses traditional mechanisms that are used by open Cloud middleware platforms.

A. CLOUDSIM

CloudSim is a Java-based discrete event simulation toolkit designed to simulate Traditional Clouds [14]. A discrete system is a system whose state variables change over time at discrete points, each of them is called an event. The tool was developed by a leading research group in Grid and Cloud computing called CLOUDS Laboratory at The University of Melbourne in Australia. The simulation tool is based on both GridSim [15] and SimJava [16] simulation tools.

CloudSim is claimed to be more effective in simulating Clouds compared to SimGrid [17] and GroudSim [18] because CloudSim allows segregation of multi-layer service (IaaS, PaaS and SaaS) abstraction [14]. This is an important feature of CloudSim that most Grid simulation tools do not support. Researchers can study each abstraction layer individually without affecting other layers.

CloudSim can be used for various goals [19]. First, it can be used to investigate the effects of algorithms of provisioning and migration of VMs on power consumption

and performance. Secondly, it can be used to test VM mechanisms that aim at allocating VMs to PMs to improve performance of VMs. It is, also, possible to investigate several ways to minimise the running costs for CSPs without violating the SLAs. Furthermore, CloudSim enables researchers to evaluate various scheduling mechanisms of tasks submitted to running VMs from the perspective of Cloud brokers. Scheduling mechanism can help in decreasing response time and thus improve performance.

Although CloudSim is considered the most mature Cloud simulation tool, the tool falls short in providing several important features. The first is that does not simulate performance variations of simulated VMs when they process tasks [19]. Secondly, service failures are not simulated in CloudSim [20]. The service failures include failures in tasks during running time and complex overhead of complicated tasks. Furthermore, CloudSim lacks the ability to simulate dynamic interaction of nodes in the infrastructure level. CloudSim allows static configuration of nodes, which remain without change during run time. Lastly, node failures are not included in CloudSim tool. DesktopCloudSim enables the simulation of dynamic nodes and node failures while performance variations and service failures are simulated by other tools. Section VI discusses those tools.

Several simulators have been published to simulate Grid computing. SimGrid [17] is one of the early simulation tools to simulate Grid environment. GridSim [15] is another tool fits within the same goal. CloudSim is built on top of GridSim. Donassole et al. [21] extended SimGrid to enable simulating Desktop Grids. Their work enables building a Grid on top of resources contributed by the public. The simulation tool is claimed to be of high flexibility and enable simulating highly heterogenous nodes. GroudSim [18] is a scalable simulation tool to simulate both Grid and Cloud platforms. The tool lets researchers to inject failures during running time. However, all of these tools fall in short to provide virtualisation feature, which is essential to evaluate VM allocation mechanisms.

MDCSim [22] is a commercial, discrete-event simulation tool developed at Pennsylvania State University to simulate multi-tier data centres and complex services in Cloud computing. It has been designed with three-level architecture, including a user-level layer, a kernel layer and communication layer for modelling the different aspects of a Cloud system. MDCSim can analyse and study a cluster-based data centre with in-depth implementation of each individual tier. The tool can help in modelling specific hardware characteristics of different components of data centres such as servers, communication links and switches. It enables researchers to estimate the throughput, response times and power consumption. However, as the simulation tool is a commercial product, it is unsuitable to run experiments.

GreenCloud [23] is another cloud simulation framework, implemented in C++ and focused on the area of power consumption and its measurement. The tool was developed on top of Ns2, a packet-level network simulation tool [24]. Having the tool implemented in C++ makes it feasible to simulate a large number of machines (100,000 or more),

while Java is assumed to be able to handle only 2GB memory on 32 bit machines. However, CloudSim was able to simulate and instantiate 100,000 machines in less than 5 minutes with only 75 MB of RAM, according to Sakellari and Loukas 2013. Although GreenCloud can support a relatively large number of servers, each may have only a single core. In addition, the tool pays no attention to virtualisation, storage and resource management.

iCanCloud [26] is a C++ based open source Cloud simulation tool based on SIMCAN [27], a tool to simulate large and complex systems. It was designed to simulate mainly IaaS Cloud systems, such as instance-based clouds like EC2 Amazon Cloud. iCanCloud offers the ability to predict the trade-off between performance and cost of applications for specific hardware to advise users about the costs involved. The tool has a GUI feature and can be adapted to different kinds of IaaS cloud scenarios. However, iCanCloud does not enable researchers to study and investigate energy efficiency solutions.

There are several extensions of CloudSim that have been developed to overcome the limitations of CloudSim tool. The extensions are NetworkCloudSim [28], WorkflowSim [20], DynamicCloudSim [19], FederatedCloudSim [29] and InterCloud [30]. NetworkCloudSim is an extension simulation tool based on CloudSim to enable the simulation of communication and messaging aspects in Cloud computing. The focus of the tool is on the network flow model for data centres and network topologies, bandwidth sharing and the network latencies involved. It also enables the simulation of complex applications such as scientific and web applications that require interconnections between them during run time. Such features can allow further accurate evaluation of scheduling and resource provisioning mechanisms in order to optimise the performance of Cloud infrastructure.

WorkflowSim is a new simulation extension that has been published recently as an extension for CloudSim tool. The tool was developed to overcome the shortage of CloudSim in simulating scientific workflow. The authors of WorkflowSim added a new management layer to deal with the overhead complex scientific computational tasks, arguing that CloudSim fails in simulating the overheads of such tasks such as queue delay, data transfer delay, clustering delay and postscripts. This issue may affect the credibility of simulation results. They also point out the importance of failure tolerant mechanisms in developing task scheduling techniques. WorkflowSim focuses on two types of failures: tasks failure and job failure. A task contains a number of jobs, so failure in a task causes a series of jobs to fail. However, our work differs from WorkflowSim in the failure event and its impact. The focus of this research is on the infrastructure level, containing nodes hosting VMs, whereas its authors were interested in the service level, that is, tasks and applications. It can be argued that service providers should consider developing failure-tolerant mechanisms to overcome such events in the infrastructure level.

DynamicCloudSim is another extension for CloudSim tool. Its authors were motivated by the fact that CloudSim lacks the ability to simulate instability and dynamic

performance changes in VMs during runtime. This can have a negative impact on the outcome of computational intensive tasks, which are quite sensitive to the behaviour of VMs. The tool can be used to evaluate scientific workflow schedulers, taking into consideration variance in VM performance. In addition, the execution time of a given task is influenced by the I/O-bound such as reading or writing data. Its authors extended instability to include task failure. Performance variation of running VMs is an open research challenge, but beyond the scope of this study.

FederatedCloudSim [29] is an extension tool in the CloudSim toolkit to enable the simulation of federated Clouds using difference federation scenarios, while respecting SLAs. According to Goiri et al. [31], Cloud Federation is the idea of bringing many CSPs together in order to avoid the case of over-demand for Cloud services by letting a CSP rent out CSPs to other computing facilities. FederatedCloudSim enables researchers to simulate and study various ways to standardise interfaces and communications between CSPs in a federated Cloud. Such a tool can help to study optimisation solutions for exchanging Cloud services between CSPs without violation of SLAs. InterCloud is another simulation tool that has been developed to simulate Cloud federation, based on the CloudSim tool. However, InterCloud falls short of providing sufficient simulation capabilities of SLAs, compared to FederatedCloudSim.

B. The Architecture of DesktopCloudSim

Simulation is necessary to investigate issues and evaluate solutions in Desktop Clouds because there is no real Desktop Cloud system available on, which to run experiments. In addition, simulation enables control of the configuration of the model to study each evaluation metric. In this research, CloudSim is extended to simulate the resource management model. CloudSim allows altering the capabilities of each host machines located in the *data centre* entity in the simulation tool. This feature is very useful for experimentations, as it is needed to set the infrastructure (i.e., physical hosts) to have an unreliable nature. This can be achieved by extending the *Cloud Resources* layer in the simulation tool. Figure 1 Depicts the layered architecture of CloudSim combined with an abstract of the DesktopCloudSim extension.

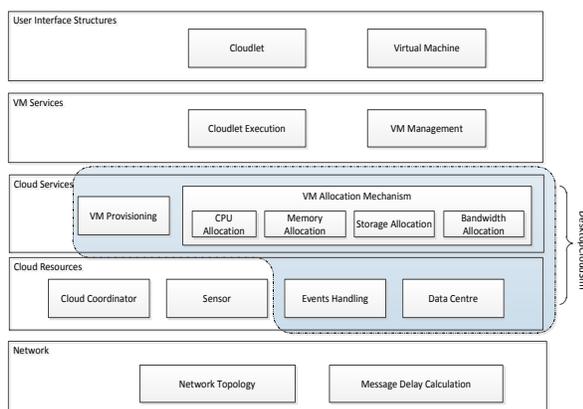


Figure 1. DesktopCloudSim Abstract

Figure 2 illustrates the components of DesktopCloudSim that read FTA trace files, as explained later in this paper. The trace files contain the failure events of PMs. The *Failure Analyser* component analyses the files of failures to send failure events to *Failure Injection* component. The *Failure Injection* component receives failure events from the *Failure Analyser* and inject failures into associated PMs during run time by sending events to *Available PMs* component. The *Available PMs* contains a list of PMs that are ready to be used, so if a PM fails then it is removed or, if a PM joins, it is added. The *Failure Injection* component informs the *VM Mechanism* unit if a PM fails, to let it restart the failed VMs on another live node or nodes. The *VM Provisioning* component provisions VMs instances to be allocated to PMs selected by *Select PM*. The *VM Mechanism* controls, which PM hosts a VM instance. The *VM Mechanism* creates restart VM instances. In addition, the *VM Mechanism* can replicate a running VM instance, if required.

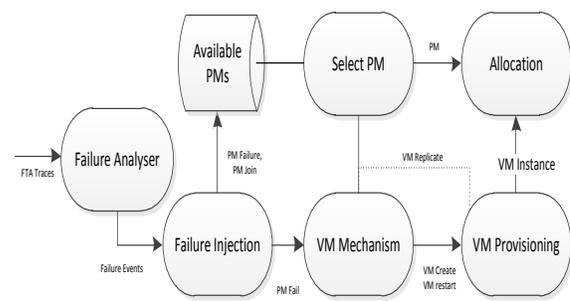


Figure 2. DesktopCloudSim Model

C. VM Allocation Mechanisms

Several VM allocation mechanisms that are employed in open Cloud platforms are discussed in this subsection. VM allocation mechanisms are: (i) Greedy mechanism, which allocates as many VMs as possible to the same PM in order to improve utilisation of resources; (ii) RoundRobin mechanism allocates the same number of VMs equally to each PM; and (iii) First Come First Serve (FCFS) mechanism allocates a requested VM to the first available PM that can accommodate it. This paper is limited to these mechanisms because they are implemented in open source Cloud management platforms such as Eucalyptus [32], OpenNebula [33] and Nimbus [34].

When a VM is requested to be instantiated and hosted to a PM, the FCFS mechanism chooses a PM with the least used resources (CPU and RAM) to host the new VM. The Greedy mechanism allocates a VM to the PM with the least number of running VMs. If the chosen PM cannot accommodate the new VM, then the next least VM running PM will be allocated. RoundRobin is an allocation mechanism, which allocates a set of VMs to each available physical host in a circular order without any priority. For example, suppose three VMs are assigned to two PMs. The RoundRobin policy will allocate VM1 to PM1 then VM2 to PM2 then allocate VM3 to PM1 again. Although these

mechanisms are simple and easy for implementation, they have been criticised for being underutilisation mechanisms, which waste energy [35]. The FCFS mechanism is expected to yield lowest throughput among the aforementioned mechanisms because it assigns VMs to PMs in somehow random manor.

IV. EXPERIMENT

The experiment is conducted to evaluate VM mechanisms mentioned in Section III.C. There are two input types needed to conduct the experiment. The first input is the trace file that contains failure events happening during the run time. Failure trace files are collected from an online archive. Subsection A discusses further this archive. The second input set is the workload submitted to the Desktop Cloud during running time. Subsection B talks about this workload.

A. Failure Trace Archive

Failure Trace Archive (FTA) is a public source containing traces of several distributed and parallel systems [36]. The archive includes a pool of traces for various distributed systems including Grid computing, Desktop Grid, peer-to-peer (P2P) and High Performance Computing (HPC). The archive contains timestamp events that are recorded regularly for each node in the targeted system. Each event has a state element that refers to the state of the associated node. For example, an event state can be unavailable, which means this node is down at the timestamp of the event. The unavailable state is considered a failure event throughout this report. The failure of a node in an FTA does not necessarily mean that this node is down. For example, a node in a Desktop Grid system can be become unavailable because its owner decides to leave the system at this time.

The Notre Dame and SETI@home FTAs were retrieved from Failure Trace Archive website. The NotreDame FTA represents an archive of a pool of heterogeneous resources that have run for 6 months within the University of Notre Dame during 2007 [37]. The nodes of this archive can be used to simulate the behaviour of nodes in a private Desktop Cloud system. Each month is provided separately representing the behaviour of nodes located in the University of Notre Dame. The FTA contains 432 nodes for month 1, 479 nodes for month 2, 503 nodes for month 3, 473 nodes for month 4, 522 nodes for month 5 and 601 nodes for month 6. The second trace archive is SETI@home FTA. The FTA has a large pool of resource (more than 200 thousand nodes) that have been run for a year in 2008/09 [38]. The nodes in SETI@home are highly heterogeneous because most of these computing nodes are denoted by the public over the Internet. A random sample of 875 nodes has been selected from SETI@home FTA for six months. The selected PMs are those who have trace files with sufficient failure events to simulate SETI@home Desktop Cloud, which is considered a public Desktop Cloud system.

We calculated the average percentage failure of nodes on every hour basis. Such study can help in evaluating the behaviour of VM mechanisms. The failure percentage is calculated as:

$$failure(h) = \frac{\text{numbr of failed nodes at } h}{\text{total number of nodes}} * 100$$

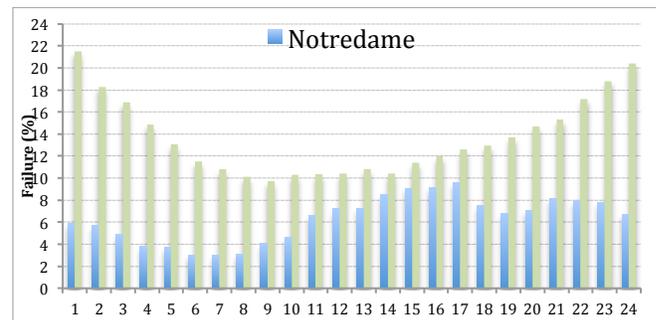


Figure 3. Average Hourly Failure

Figure 3 shows an average hourly failure percentage in 24 hour-period for analysis of 6 months run times of NotreDame and SETI@home nodes. The period is set to 24 hours because this is the running time set for our experiments. NotreDame failure analysis shows that failure percentage is about 3% as minimum in hour 6. Hour 17 recorded the highest failure percentages at about 10%. It is worth mentioning that on average about 6.3% of running nodes failed in an hour during the 6-month period. For SETI@home nodes, the highest failure percentage was about 21% in hour 1 while lowest was about 10%. However, it was recorded that the percentage of node failures can reach up to 80% in some hours. Overall, the average hourly failure rate of SETI@home is about 13.7%. This can demonstrate that failure events in Desktop Clouds are norms rather than exceptions.

B. Experiment Setting

The experiment is run for 180 times once for NotreDame Desktop Cloud and another for SETI@home Desktop Cloud, each time represent a simulation of running NotreDame Desktop Cloud for one day. The run time set to one day because the FTA provides a daily trace for NotreDame nodes as mentioned above. Each VM allocation mechanism is run for 180 times representing traces of 6 months from the FTA. This makes the total number of runs is 540 (3 * 180). The workload was collected from the PlanetLab archive. The archive provides traces of real live applications submitted to the PlanetLab infrastructure [39]. One day workload was retrieved randomly as input data in this experiment. Each task in the workload is simulated as a Cloudlet in the simulation tool. The workload input remains the same during all the experiment runs because the aim of this experiment is to study the impact of node failures on throughput of Desktop Clouds.

The FTA files provide the list of nodes along with timestamps of failure/alive times. However, the specifications of nodes are missing. Therefore, we had set specification up randomly for physical machines. The missing specifications are technical specifications such as CPU power, RAM size and hard disk size.

Clients requested that 700 instances of VMs to run for 24 hours. There are four types of VM instances: *micro*, *small*, *medium* and *large*. They are similar to VM types that are offered by Amazon EC2. The type of each requested VM instance is randomly selected. The number of requested VMs and types remain the same for all run experiment sets. Each VM instance receives a series of tasks to process for a given workload. The workload is collected from PlanetLab archive, which is an archive containing traces. PlanetLab is a research platform that allows academics to access a collection of machines distributed around the globe. A one day workload of tasks was collected using CoMon monitoring tool [40]. The same workload is submitted in every one day run.

In the experiment, if a node fails then all hosted VMs will be destroyed. The destruction of a VM causes all running tasks on the VM to be lost, which consequently affect the throughput. The lost VM is started again on another PM and begins receiving new tasks. During running time, a node can become alive and re-join the Cloud according to the used failure trace file. The simulation was run on a Mac i27 (CPU = 2.7 GHz Intel Core i5, 8 GB MHz DDR3) running OS X 10.9.4. The results were analysed using IBM SPSS Statistics v21 software.

V. RESULTS AND DISCUSSION

Table II shows a summary of descriptive results obtained when measuring the throughput output for each VM allocation mechanism implemented in NotreDame Cloud. Kolmogorov-Smirnov (K-S) test of normality shows that the normality assumption was not satisfied because the FCFS and Greedy mechanisms are significantly non-normal, $P < 0.05$. Therefore, the non-parametric test Friedman's ANOVA was used to test which mechanism can yield better throughput. Friedman's ANOVA test confirms that throughput varies significantly from mechanism to another, $X_F^2(2) = 276.6, P < 0.001$. Mean, median, variance and standard deviation are reported in Table II.

Table II. Throughput Results for NotreDame Desktop Cloud

Mechanism	Mean (%)	Median (%)	Var.	St. Dev.	K-S Test
FCFS	82.66	82.2	40.32	6.35	P = 0.034
Greedy	92.47	93.1	18.34	4.28	P < 0.001
RoundRobin	89.14	89	16.47	4.06	P = 0.2

Three Wilcoxon pairwise comparison tests were conducted to find out which mechanism with highest throughput. Note that three tests are required to compare three pairs of mechanisms, which are FCFS Vs. Greedy, FCFS Vs. RoundRobin and Greedy Vs. RoundRobin mechanisms. The level of significance was altered to be 0.017 using Bonferroni correction [41] method because there were 3 post-hoc tests required ($0.05/3 \approx 0.017$). The tests show that there is a significant different between each mechanism with its counterpart. Therefore, it can be concluded that Greedy mechanism yield highest throughput

since it has the median with highest value (median = 92.47%).

The median throughput of FCFS was about 83%, as being the worst mechanism among the tested mechanisms. The RoundRobin came second in terms of throughput because the mechanism distributes load equally. So, node failures are ensured to affect the throughput. The median throughput was about 92% when Greedy VM mechanism was employed. The mechanism aims at maximising utilisation by packing as many VMs as possible to the same PM, thus reduce the number of running PMs. The average failure rate in submitted tasks is about 8%, given the average node failure percentage is about 6% as Section IV.A shows.

Table III shows a summary of descriptive results obtained for throughput output for the FCFS, Greedy and RoundRobin VM allocation mechanisms employed in SETI@home Desktop Cloud. Kolmogorov-Smirnov (K-S) test of normality shows that the normality assumption was violated because the FCFS and RoundRobin mechanisms are significantly non-normal, $P < 0.05$. Therefore, the non-parametric test Friedman's ANOVA was used to test, which mechanism can yield better throughput. Friedman's ANOVA test confirms that throughput varies significantly from mechanism to another, $X_F^2(2) = 86.63, P < 0.001$. Mean, median, variance and standard deviation are reported in Table III.

Table III. Throughput Results for SETI@home Desktop Cloud

Mechanism	Mean (%)	Median (%)	Var.	St. Dev.	K-S Test
FCFS	82.04	83.28	20.23	4.5	P < 0.001
Greedy	81.80	81.93	16.1	4.01	P = 0.2
RoundRobin	80.45	81.04	16.11	4.01	P = 0.004

Three Wilcoxon pairwise comparison tests were conducted to find out which mechanism yielded highest throughput. As explained before, three tests are required to compare three pairs of mechanisms which are FCFS Vs. Greedy, FCFS Vs. RoundRobin and Greedy Vs. RoundRobin mechanisms. The level of significance was altered to be 0.017 using Bonferroni correction [41] method because there were 3 post-hoc tests required ($0.05/3 \approx 0.017$). The tests show that there is a statistically significant different between RoundRobin vs. Greedy mechanisms and RoundRobin vs. FCFS mechanisms. However, Greedy vs. FCFS mechanisms did not show a significant difference. Therefore, it can be which mechanisms yielded highest throughput.

The throughput results of employed mechanism for SETI@home Desktop Cloud showed that the difference between throughput of results were quite limited, by less than 2%. The FCFS and Greedy mechanisms yielded highest throughput at about 82% and 81% respectively. RoundRobin came the last with throughput of about 80% only. The mean reason behind the drop of throughput results of mechanisms in SETI@home Desktop Cloud compared to NotreDame Desktop Cloud is the average failure rate of nodes in

SETI@home is almost the double of average failure rate of NotreDame nodes. We can conclude that based on the results of experiments there is a potential to develop fault-tolerant VM mechanism for Desktop Cloud systems.

VI. RELATED WORK

The literature shows that the focus is on how to minimise the power consumed by physical nodes in order to maximise revenue for CSPs. Researchers are motivated to tackle the issue because power in data centres accounts for a large proportion of maintenance costs [42]. The idea is that better utilisation leads to more servers that are idle, so can be switched to power saving mode (e.g., sleep, hibernation) to reduce their energy consumption. According to Kusic et al. an idle machine uses as much as 70% of the total power consumed when it is fully utilised [9].

Srikantiah et al. studied the relationship between energy consumption, resource utilisation and performance in resource consolidation in Traditional Clouds [43]. The researchers investigated the impact of resource high utilisation on performance degradation when various VMs are consolidated at the same physical node, introducing the notion of optimal points. They argued that there is a utilisation point that allows placement of several VMs at the same physical node without affecting performance. Once this point is reached in a PM, no new VMs are placed, and the proposal is to calculate this optimal point of utilisation then to employ a heuristic algorithm for VM placement, since the authors defined the consolidation problem as a multi-dimensional Bin Packing problem and showed that the consumption of power per transaction results in a 'U'-shaped curve. They found that CPU utilisation at 70% was the optimal point in their experiment, but that it varied according to the specification of the PMs and workload. The approach is criticised because the technique adopted depends heavily on the type of the workload and the nature of the targeted machines [44].

Verma et al. presented 'pMapper', a power-aware framework for VM placement and migration in virtualised systems, where the monitoring engine collects current performance and power status for VMs and PMs in case migration is required [45]. The allocation policy in pMapper employs *mPP*, an algorithm that places VMs on servers with the aim of reducing the power they consume. The algorithm has two phases. The first is to determine a target utilisation point for each available server based on their power model. The second is to employ a First Fit Decreasing (FFD) heuristic solution to place VMs on servers with regard to the utilisation point of each. The optimisation in the framework considers reducing the cost of VM migration from one server to another. The migration cost is calculated by a migration manager for each candidate PM in order to determine which node is chosen. The work is criticised as it does not strictly comply with SLA requirements [46]; the proposed allocation policy deals with static VM allocation where specifications of VMs remain unchanged. This is not the case in Cloud computing, where clients can scale up or down dynamically. In addition, it requires prior knowledge of each PM in order to compute the power model.

Meng et al. proposed a VM provisioning approach to consolidate multiple VM instances for the same PM in order to improve resource utilisation and thus reduce the energy consumed by under-utilised PMs [47]. A VM selection algorithm was developed to identify compatible VM instances for consolidation. Compatible VM instances are those with similar capacity demand, defined as their application performance requirement, and these are grouped into sets allocated to the minimum number of PMs. It can be argued that consolidating compatible VM instances to the same PMs will have a small negative effect on applications assigned to each VM instance and thus keep SLA requirements from being violated. The study found an improvement of 45% in resource utilisation.

The authors in [48] and [32] devised an algorithm to allocate VM instances to PMs at data centres with the goal of reducing power consumption in PMs without violating the SLA agreement between a Cloud provider and users. The researchers argued that assigning a group of VMs to as few PMs as possible will save power [49]. The energy-aware resource algorithm [46] has two stages: VM placement and VM optimisation. The VM placement technique aims to allocate VMs to PMs using a Modified Best Fit Decreasing (MBFD) algorithm. This is based on the Best Fit Decreasing (BFD) algorithm that uses no more than $11/9 * OPT + 1$ bins (OPT is the optimal number of bins) [50].

The MBFD algorithm sorts VMs into descending order of CPU utilisation in order to choose power-efficient nodes first. The second stage is the optimisation step responsible for migrating VMs from PMs that are either over- or under-utilised. However, VM migration may cause unwanted overheads, so should be avoided unless doing so reduces either power consumption or performance, so the authors set lower and upper thresholds for utilisation. If the total utilisation of the CPU of a PMs falls below the lower threshold, this indicates that the host might consume more energy than it needs. Similarly, if the utilisation exceeds the upper threshold then the performance of the hosted VMs may deteriorate. In this case, some VMs should migrate to another node to reduce the level of utilisation. The authors concluded that the Minimisation of Migrations (MM) policy could save up to 66% of energy, with performance degradation of up to 5%. It was found that the MM policy minimised the number of VMs that have to migrate from a host in the event of utilisation above the upper threshold.

Graubner et al. proposed a VM consolidation mechanism based on a live migration technique with the aim of saving power in Cloud computing [44]. They developed a relocation algorithm that periodically scans available PMs to determine which PM to migrate VM instances from, and which PM to migrate them to. The approach was found to save up to 16% of power when implemented in the Eucalyptus platform, however the relocation process was unclear, with no further explanation of when it is triggered during run time [51].

The authors in [52] proposed GreenMap, a power-saving VM-based management framework under the constraint of multi-dimensional resource consumption in clusters and data centres. GreenMap dynamically allocates and reallocates VMs to a set of PMs within a cluster during runtime. There

are four modules in the framework: clearing; locking; trade-off; and placement. The clearing module is responsible for excluding VMs inappropriate for dynamic placement, for instance those with unpredictable or rapid variation in demand. The locking module monitors SLA violations caused by the workload, in which event the module will switch to a redundant VM for execution. The trade-off module evaluates the potential of a new placement generated by the placement module in respect of performance and cost trade-off. The placement module performs a strategy for reallocating live VMs to another physical resource to save power, based on a configuration algorithm. The algorithm starts by randomly generating a new placement configuration. The placement module then delivers the configuration to the trade-off module. The experiment showed that it is possible to save up to 69% of power in a cluster, with some performance degradation, but it did not consider the overheads of the placement module.

The authors in [51] proposed an energy-saving mechanism developed and implemented for a private Cloud called Snooze, tested using a dynamic web workload. The authors argued that it differed from other power-aware VM mechanisms in two aspects, in that it was applied and tested in a realistic Cloud environment, and that it takes dynamic workload into consideration. A monitor unit was introduced periodically to check running PM; any under- or over-utilised nodes were reported to a general manager module to issue a migration command. There are four VM allocation policies: placement; overload relocation; underload relocation; and consolidation.

The placement policy allocates new VM instance requests to PMs using RoundRobin scheduling, which distributes the load to PMs in a balanced way. The overload policy scans PMs to check if a PM is overloaded with VM instances and, if so, searches for a PM that is only moderately loaded to accommodate these VM instances in all-or-nothing way (i.e., migrate all running VMs or none). The migration command is sent to the migration policy for straightforward execution. Similarly, the underload policy issues a migration command to migrate VMs from under-utilised PM in an all-or-nothing way. The mechanism managed to save up to 60% of power, the experiment concluded, but it was conducted in a homogenous infrastructure, that is, it assumed that all PMs have the same computing capacity. In addition, the all-or-nothing method may be a drawback as it leads to PMs being overloaded, which may cause performance degradation in instances of hosted VM.

Van et al. proposed a virtual resource manager focused on maintaining service levels while improving resources utilisation via a dynamic placement mechanism [53]. The manager has two levels: a local decision module and a global decision module. The first is concerned with applications, as the manager deals with complex N-tier levels in, for instance, online applications that require more than one VM instance to process. The global decision module has two stages: the VM placement stage, concerned with allocating a VM to a specific PM with the goal of improving resource utilisation; and the VM provisioning stage of scheduling

applications to VMs (i.e., sending applications to be processed by VM instances).

The authors in [54] proposed a novel VM placement approach of two phases: candidacy and placement. The former elects a list of PMs eligible to accommodate VM instances, choosing the candidate PM on the basis of migration capability, network bandwidth connectivity and user deployment desire, which should be available beforehand. Available PMs have a four-level hierarchy representing an ordering system of PMs available to be candidates. The latter phase selects one of the candidate PMs from the first phase to host a VM instance on the basis of low-level constraints. The authors argue that the first phase can help to reduce the time spent choosing the most suitable PM. However, this work requires prior knowledge of user deployment of VM instances, which is not supported in CSPs. CSPs usually offer different classes of VM instances for end users to choose between. Asking further questions regarding user preferences is not economically viable.

The authors in [55] proposed a VM placement technique that employs the FF heuristic solution to maximise revenue for CSPs under performance constraints, expressed as an SLA violation metric measuring performance degradation of VM instances caused by using the FF mechanism to improve resource utilisation. The proposed system has two managers: the global manager decides which PM hosts a VM instance; and the local manager is concerned with scheduling VM instances within the hosted PM. The global manager employs a decision-making policy for each candidate PM's viability for hosting a VM instance in such a way as to improve resource utilisation.

Calcavecchia et al. proposed the Backward Speculative Placement as a novel VM placement technique [56]. The VM placement technique has two phases: continuous deployment and ongoing optimisation. The continuous deployment phase allocates a VM instance to the PM with the highest demand risk, a scoring function to measure the level of dissatisfaction with a PM at the final unit of time. It is, however, not clearly explained how this is awarded. The ongoing optimisation phase migrates VM instances hosted to a PM with high risk demand to another PM with a low score, as long as it is able to accommodate the VM instances. The Backward Speculative Placement technique was able to decrease the execution time of submitted tasks.

The authors in [57] proposed a VM placement and migration approach to minimise the effect of transfer time of data between VM instances and data storage. In Cloud computing, a CSP can provide VM instances to end users to process data while these data are stored in different locations, for example Amazon EC2 and Amazon S3. Therefore, the approach developed takes network I/O requirements into consideration when VM placement is applied. In addition, the VM migration policy is triggered when the time required to transfer data exceeds a certain threshold. Network instability is the main reason for this increase of time, and the threshold is stated in the SLA agreement. The study showed that the time taken to complete the task fell, on average, due to the placement of VM, depending on location.

A novel traffic-aware VM placement technique was developed by [58] with the goal of improving network scalability. The mechanism employs a two-tier approximate algorithm to place VM instances with PMs in such a way that significantly reduces the aggregate traffic in datacentres. The two-tier algorithm partitions VMs and PMs separately into clusters. The VMs and PMs are matched individually in each cluster. The partitioning step is achieved using a classical min-cut graph algorithm that assigns each VM pair with a high mutual traffic rate to the same VM cluster. Having VM instances with a high traffic rate in the same cluster of PMs means that traffic is exchanged only through that cluster, which can reduce the load upon switches at a data centre.

Purlieus [59] is a resource allocation tool developed to improve the performance of MapReduce jobs and to reduce network traffic by paying attention to the location of resources. MapReduce enables the analysis and processing of large amount of data in a quick and easy way [60]. Purlieus employs VM placement techniques that allocate VM instances to PMs according to their location. Purlieus was able to reduce the execution time of jobs by 50% for a variety of types of workload.

The authors in [61] studied the VM allocation problem from the network perspective [61]. They proposed a novel VM placement mechanism that considers network constraint, which is the variation in traffic demand time. Its goal is to minimise the load ratio across all network cuts by implementing a novel mechanism, the two-phase connected component-based recursive split, to choose the PM with which to place a VM instance. It exploits the recursive programming technique to formulate a ranking table of each VM instance that is connected. The PM with the least connected ranks of associated VMs is selected to host a new VM instance, but the proposed mechanism is for static VM placement only, thus it does not consider moving VM instances around during run time to reduce the cut load ratio.

The authors in [62] introduced S-CORE, a scalable VM migration mechanism to reallocate VM instances to PMs dynamically with the goal of minimising traffic within a datacentre. They showed that S-CORE can achieve cost reductions in communication of up to 80% with a limited amount of VM migration. S-CORE assigns a weight for each link in a datacentre, taking into consideration the amount of data traffic routed over these links. If the line weight exceeds a certain threshold, then some VM instances with high traffic load have to migrate to another PM using a different link. Such an approach avoids traffic congestion on core links at data centres to prevent any degradation in the performance of a Cloud system.

The aforementioned studies investigated various VM allocation mechanisms with the aim of minimising power consumption, improving performance or reducing the traffic load in Cloud systems. However, they all fell short of providing a mechanism tolerant of failure events in Clouds' PMs. Therefore, these VM allocation techniques are neither practical to employ nor to implement in a Desktop Cloud system. The following subsection reviews several studies that have tackled the issue of node failure.

A wide range of techniques and approaches has been developed to tackle node failure issues in Desktop Grid systems, because a node within a Desktop Grid system can voluntarily join or leave the system, increasing the probability of node failure, heightening the risk of losing results. For example, the authors in [63] developed a fault-tolerant technique in Desktop Grid systems that employs replication of applications to avoid losing them in failure events. Another approach was proposed by [64], based on the mechanism of application migration. This checks applications periodically during runtime, and in the event of node failures all associated application are restored and migrated to another node. However, this is not practical in this study because it is concerned with the applications level and violates the concept of the Cloud computing paradigm that isolates the infrastructure layer from the service layer to prevent CSPs from having control over services run by end users.

Machida et al. proposed a redundancy technique for server consolidation [65]. The focus was complex online applications requiring several VM instance for each application, and the technique offers k fault tolerance with the minimum number of physical servers required for application redundancy [65]. It relies on replicating an application a times and running it for k number of VM instances. The number of VM instances is calculated on the basis of the requirements of application a , but requires full knowledge of and access to the applications and services that run on VM instances in order to replicate them. This, again, violates the concept of Cloud computing whereby CSPs are prevented from being able to access and control the applications of end users. Furthermore, the approach assumes that all physical servers have the same computing capacity, impractical in the era of Cloud computing where PMs are usually quite heterogeneous.

The authors in [66] proposed the BFTCloud, a fault-tolerant framework for Desktop Cloud systems that tackles the specific malicious behaviour of nodes known as Byzantine faults: machines that provide deliberately wrong results. The framework employs a replication technique with a primary node by $3 * f$, where f is the number of faulty nodes at run time. The framework considers failure probability as the mean to choose primary nodes and their replicas in respect of QoS requirements. Byzantine faults are identified by comparing the results reported by a primary node with those of its replica; if the results are inconsistent then they will be sent to another node to process and compared to detect which machine is behaving suspiciously. However, the calculation of failure probability is not clearly given. In addition, although the framework was said to be for Desktop Cloud systems, it does not possess the essential feature of employing virtualisation to keep the service layer isolated from the physical layer; in fact, the technique is to replicate tasks by sending one to a primary node and its $3 * f$ replicas of nodes. Another issue worth mentioning about the BFTCloud mechanism is the notion of f , which means that the number of faulty nodes should be known before run time. However, this technique is impractical since the number of

node failures in such distributed systems is unpredictable and difficult to calculate [67].

The authors in [68] addressed the issue of node failure in hybrid Clouds, that is, private and public Clouds. The problem is formulated as follows: a private Cloud with limited resources (i.e., PMs) has a certain number of nodes with a high failure rate. The question is how to minimise the dependency of public Clouds to achieve better QoS, given that sending workload to a public Cloud costs more. The authors proposed a failure-aware VM provisioning for hybrid Clouds, a ‘time-based brokering strategy’, to handle failure of nodes in private Clouds by redirecting tasks required long term into a public Cloud. The decision to forward a task to a public Cloud is based on the duration of the request; if longer than the mean request duration of all tasks, then it will be forwarded. Although the proposed strategy considers that a public Cloud solves the issue of node failure in private Clouds, the issue is not answered unless the reliability of this public Cloud can be guaranteed.

The review of VM mechanisms in this section shows that the design of a fault-tolerant VM allocation mechanism remains an open research problem that needs to be tackled in Cloud environments with faults, such as in Desktop Cloud systems.

VII. CONCLUSION

Desktop Cloud can be seen as a new direction in Cloud computing. Desktop Cloud systems exploit idle computing resources to provide Cloud services mainly for research purposes. The success of Desktop Grids in providing Grid capabilities stimulated the concept of applying the same concept within Cloud computing. However, Desktop Clouds use infrastructure that is very volatile since computing nodes have high probability to fail. Such failures can be problematic and cause negative on the throughput of Desktop Clouds.

This paper presented a DesktopCloudSim as an extension tool CloudSim, a widely used Cloud simulation tool. DesktopCloudSim enables the simulation of node failures in the infrastructure of Cloud. We demonstrated that the tool can be used to study the throughput of a Desktop Cloud using NotreDame and SETI@home FTA traces. We showed that the average failure rate of nodes in NotreDame and SETI@home FTAs. Such study can help to show that node failure in Desktop Cloud is quietly expected.

The results of experiments demonstrate that node failures affect negatively the throughput outcome of Desktop Clouds. However, the related works lack the ability to solve the problem of throughput decrease as a result of node failures.

This opens a new direction to design a fault tolerant mechanism for Desktop Cloud. We intend to develop such mechanism and evaluate it using the proposed tool. In addition, several metrics such as power consumption and response time should be used to evaluate VM mechanism.

REFERENCES

[1] A. Alwabel, R. Walters, and G. B. Wills, “DesktopCloudSim : Simulation of Node Failures in The Cloud,” in *The Sixth*

International Conference on Cloud Computing, GRIDs, and Virtualization CLOUD COMPUTING 2015, pp. 14-19, 2015.

[2] R. Buyya, R. Buyya, C. S. Yeo, C. S. Yeo, S. Venugopal, S. Venugopal, J. Broberg, J. Broberg, I. Brandic, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Futur. Gener. Comput. Syst.*, vol. 25, no. 6, p. 17, Jun. 2009.

[3] B. M. Segal, P. Buncic, D. G. Quintas, D. L. Gonzalez, A. Harutyunyan, J. Rantala, and D. Weir, “Building a volunteer cloud,” *Memorias la ULA*, 2009.

[4] A. Alwabel, R. Walters, and G. B. Wills, “A View at Desktop Clouds,” in *Proceedings of the International Workshop on Emerging Software as a Service and Analytics*, 2014, pp. 55–61.

[5] G. Kirby, A. Dearle, A. Macdonald, and A. Fernandes, “An Approach to Ad hoc Cloud Computing,” *Arxiv Prepr. arXiv1002.4738*, 2010.

[6] A. Marinos and G. Briscoe, “Community Cloud Computing,” pp. 472–484, 2009.

[7] A. Andrzejak, D. Kondo, and D. P. Anderson, “Exploiting non-dedicated resources for cloud computing,” in *Proceedings of the 2010 IEEE/IFIP Network Operations and Management Symposium, NOMS 2010*, 2010, pp. 341–348.

[8] A. Gupta and L. K. L. Awasthi, “Peer enterprises: A viable alternative to Cloud computing?,” in *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on*, 2009, vol. 2, pp. 1–6.

[9] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, “Power and performance management of virtualized computing environments via lookahead control,” *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, Oct. 2008.

[10] A. Marosi, J. Kovács, and P. Kacsuk, “Towards a volunteer cloud system,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1442–1451, Mar. 2013.

[11] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The characteristics of cloud computing,” in *Proceedings of the International Conference on Parallel Processing Workshops*, 2010, pp. 275–279.

[12] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-degree compared,” in *Grid Computing Environments Workshop, GCE 2008*, 2008, pp. 1–10.

[13] S. Choi, H. Kim, E. Byun, M. Baik, S. Kim, C. Park, and C. Hwang, “Characterizing and classifying desktop grid,” in *Proceedings - Seventh IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2007*, 2007, pp. 743–748.

[14] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities,” *High Perform. Comput. Simulation, 2009. HPSCS’09*, pp. 1–11, Jun. 2009.

[15] R. Buyya and M. Murshed, “Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing,” *Concurr. Comput. Pract. and Exp.*, vol. 14, no. 13–15, pp. 1175–1220, Nov. 2003.

[16] F. Howell and R. McNab, “SimJava: A discrete event simulation library for java,” *Simul. Ser.*, 1998.

[17] H. Casanova, “Simgrid: a toolkit for the simulation of application scheduling,” *Proc. First IEEE/ACM Int. Symp. Clust. Comput. Grid*, pp. 430–437, 2001.

[18] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, “GroudSim: an event-based simulation framework for

- computational grids and clouds,” *Euro-Par 2010 Parallel Processing Workshop*, no. 261585, pp. 305–313, 2011.
- [19] M. Bux and U. Leser, “DynamicCloudSim: simulating heterogeneity in computational clouds,” *SWEET '13 Proc. 2nd ACM SIGMOD Work. Scalable Work. Exec. Engines Technol.*, 2013.
- [20] W. Chen and M. Rey, “WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments,” 2012.
- [21] B. Donassolo, H. Casanova, A. Legrand, and P. Velho, “Fast and scalable simulation of volunteer computing systems using SimGrid,” in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing - HPDC '10*, 2010, p. 605.
- [22] S. H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, “MDCSim: A multi-tier data center simulation platform,” in *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on*, 2009.
- [23] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan, “GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers,” in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, 2010, vol. 62, no. 3, pp. 1–5.
- [24] S. McCanne, S. Floyd, K. Fall, and K. Varadhan, “Network simulator ns-2.” 1997.
- [25] G. Sakellari and G. Loukas, “A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing,” *Simul. Model. Pract. Theory*, vol. 39, pp. 92–103, Dec. 2013.
- [26] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, “ICanCloud: A Flexible and Scalable Cloud Infrastructure Simulator,” *J. Grid Comput.*, vol. 10, no. 1, pp. 185–209, Apr. 2012.
- [27] A. Núñez, J. Fernández, J. D. García, L. Prada, and J. Carretero, “SIMCAN: A SIMulator Framework for Computer Architectures and Storage Networks,” in *Simutools '08 Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008.
- [28] S. K. Garg and R. Buyya, “NetworkCloudSim: Modelling parallel applications in cloud simulations,” in *Proceedings - 2011 4th IEEE International Conference on Utility and Cloud Computing, UCC 2011*, 2011, no. Vm, pp. 105–113.
- [29] A. Kohne, M. Spohr, L. Nagel, and O. Spinczyk, “FederatedCloudSim: a SLA-aware federated cloud simulation framework,” in *Proceedings of the 2nd International Workshop on CrossCloud Systems - CCB '14*, 2014, pp. 1–5.
- [30] R. Buyya, R. Ranjan, and R. N. Calheiros, “InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services,” in *Algorithms and architectures for parallel processing*, 2010, vol. 6081 LNCS, no. PART 1, pp. 13–31.
- [31] Í. Goiri, J. Guitart, and J. Torres, “Characterizing cloud federation for enhancing providers’ profit,” in *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, 2010, pp. 123–130.
- [32] D. Nurmi, R. Wolski, C. Grzegorzcyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, “The eucalyptus open-source cloud-computing system,” in *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009*, 2009, pp. 124–131.
- [33] J. Fontán, T. Vázquez, L. Gonzalez, R. S. Montero, and I. M. Llorente, “OpenNEbula: The open source virtual machine manager for cluster computing,” in *Open Source Grid and Cluster Software Conference - Book of Abstracts*.
- [34] B. Sotomayor, R. R. S. Montero, I. M. Llorente, and I. Foster, “Virtual infrastructure management in private and hybrid clouds,” *IEEE Internet Comput.*, vol. 13, no. 5, pp. 14–22, Sep. 2009.
- [35] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers,” *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.
- [36] B. Javadi, D. Kondo, A. Iosup, and D. Epema, “The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems,” *J. Parallel Distrib. Comput.*, vol. 73, no. 8, pp. 1208–1223, Aug. 2013.
- [37] B. Rood and M. J. Lewis, “Multi-state grid resource availability characterization,” *2007 8th IEEE/ACM Int. Conf. Grid Comput.*, pp. 42–49, Sep. 2007.
- [38] B. Javadi, D. Kondo, J.-M. Vincent, and D. P. Anderson, “Mining for statistical models of availability in large-scale distributed systems: An empirical study of SETI@home,” *2009 IEEE Int. Symp. Model. Anal. Simul. Comput. Telecommun. Syst.*, pp. 1 – 10, 2009.
- [39] L. Peterson, S. Muir, T. Roscoe, and A. Klingaman, “PlanetLab Architecture: An Overview,” no. May, 2006.
- [40] K. Park and V. S. Pai, “CoMon,” *ACM SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, p. 65, Jan. 2006.
- [41] A. Field, *Discovering statistics using SPSS*, Third. SAGE Publications Ltd, 2009.
- [42] L. A. Barroso and U. Hözlze, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, vol. 4, no. 1. 2009.
- [43] S. Srikantaiah, A. Kansal, and F. Zhao, “Energy aware consolidation for cloud computing,” in *HotPower'08: Proceedings of the 2008 conference on Power aware computing and systems*, 2008.
- [44] P. Graubner, M. Schmidt, and B. Freisleben, “Energy-efficient management of virtual machines in Eucalyptus,” in *2011 IEEE 4th International Conference on Cloud Computing*, 2011, pp. 243–250.
- [45] A. Verma, P. Ahuja, and A. Neogi, “pMapper: power and migration cost aware application placement in virtualized systems,” in *Middleware '08 Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, 2008, pp. 243–264.
- [46] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.
- [47] X. Meng, C. Isci, J. O. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, “Efficient resource provisioning in compute clouds via VM multiplexing,” *Proceeding 7th Int. Conf. Auton. Comput. - ICAC'10*, p. 11, 2010.
- [48] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010, pp. 826–831.

- [49] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges," *arXiv Prepr. arXiv1006.0308*, no. Vm, pp. 1–12, 2010.
- [50] V. Vazirani, *Approximation algorithms*, Second. New York, New York, USA: Springer, 2003.
- [51] E. Feller, C. Rohr, D. Margery, and C. Morin, "Energy management in IaaS clouds: A holistic approach," in *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, 2012, pp. 204–212.
- [52] X. Liao, H. Jin, and H. Liu, "Towards a green cluster through dynamic remapping of virtual machines," *Futur. Gener. Comput. Syst.*, vol. 28, no. 2, pp. 469–477, Feb. 2012.
- [53] H. N. Van, F. D. Tran, and J. M. Menaud, "SLA-aware virtual resource management for cloud infrastructures," in *Proceedings - IEEE 9th International Conference on Computer and Information Technology, CIT 2009*, 2009, vol. 1, pp. 357–362.
- [54] K. Tsakalozos, M. Roussopoulos, and A. Delis, "VM placement in non-homogeneous IaaS-clouds," in *Service-Oriented Computing*, vol. 7084 LNCS, G. Kappel, Z. Maamar, and H. R. Motahari-Nezhad, Eds. Springer Berlin Heidelberg, 2011, pp. 172–187.
- [55] W. Shi and B. Hong, "Towards Profitable Virtual Machine Placement in the Data Center," *2011 Fourth IEEE Int. Conf. Util. Cloud Comput.*, pp. 138–145, Dec. 2011.
- [56] N. M. Calcavecchia, O. Biran, E. Hadad, and Y. Moatti, "VM placement strategies for cloud scenarios," in *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, 2012, pp. 852–859.
- [57] J. T. Piao and J. Yan, "A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing," *2010 Ninth Int. Conf. Grid Cloud Comput.*, pp. 87–92, Nov. 2010.
- [58] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proceedings - IEEE INFOCOM*, 2010.
- [59] B. Palanisamy, A. Singh, L. Liu, and B. Jain, "Purlieus: Locality-aware resource allocation for MapReduce in a cloud," *2011 Int. Conf. High Perform. Comput. Networking, Storage Anal.*, pp. 1–11, 2011.
- [60] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *Proceedings of 6th Symposium on Operating Systems Design and Implementation*, 2004, vol. 103, no. 34, pp. 137–149.
- [61] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware VM placement for cloud systems," in *Proceedings - 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012*, 2012, pp. 498–506.
- [62] F. P. Tso, G. Hamilton, K. Oikonomou, and D. P. Pezaros, "Implementing scalable, network-aware virtual machine migration for cloud data centers," in *IEEE International Conference on Cloud Computing, CLOUD*, 2013, pp. 557–564.
- [63] A. Litke, D. Skoutas, K. Tserpes, and T. Varvarigou, "Efficient task replication and management for adaptive fault tolerance in Mobile Grid environments," *Futur. Gener. Comput. Syst.*, vol. 23, no. 2, pp. 163–178, 2007.
- [64] H. L. H. Lee, D. P. D. Park, M. H. M. Hong, S.-S. Y. S.-S. Yeo, S. K. S. Kim, and S. K. S. Kim, "A Resource Management System for Fault Tolerance in Grid Computing," *2009 Int. Conf. Comput. Sci. Eng.*, vol. 2, pp. 609–614, 2009.
- [65] F. Machida, M. Kawato, and Y. Maeno, "Redundant virtual machine placement for fault-tolerant consolidated server clusters," in *Proceedings of the 2010 IEEE/IFIP Network Operations and Management Symposium, NOMS 2010*, 2010, pp. 32–39.
- [66] Y. Zhang, Z. Zheng, and M. R. Lyu, "BFTCloud: A Byzantine Fault Tolerance Framework for Voluntary-Resource Cloud Computing," *2011 IEEE 4th Int. Conf. Cloud Comput.*, pp. 444–451, Jul. 2011.
- [67] B. Javadi, D. Kondo, J. Vincent, and D. P. Anderson, "Discovering Statistical Models of Availability in Large Distributed Systems: An Empirical Study of SETI@home," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1896–1903, 2011.
- [68] B. Javadi, J. Abawajy, and R. Buyya, "Failure-aware resource provisioning for hybrid Cloud infrastructure," *J. Parallel Distrib. Comput.*, vol. 72, no. 10, pp. 1318–1331, Oct. 2012.

Semantic Indexing based on Focus of Attention Extended

by Weakly Supervised Learning

Kimiaki Shirahama*, Tadashi Matsumura†, Marcin Grzegorzec*, and Kuniaki Uehara†

* Pattern Recognition Group, University of Siegen

† Graduate School of System Informatics, Kobe University

Email: kimiaki.shirahama@uni-siegen.de, tadashi@ai.cs.kobe-u.ac.jp,
marcin.grzegorzec@uni-siegen.de, uehara@kobe-u.ac.jp

Abstract—Semantic Indexing (SIN) is the task to detect concepts like *Person* and *Car* in video shots. One main obstacle in SIN is the abundant information contained in a shot where not only a target concept to be detected but also many other concepts are displayed. In consequence, the detection of the target concept is adversely affected by other irrelevant concepts. To overcome this, we enhance SIN based on a human brain mechanism to effectively select important regions in the shot. Specifically, SIN is integrated with Focus of Attention (FoA) which identifies salient regions that attract user’s attention. The feature of a shot is extracted by weighting regions based on their saliencies, so as to suppress effects of irrelevant regions and emphasise the region of the target concept. In this integration, it is laborious to prepare salient region annotation that assists detecting salient regions most likely to contain the target concept. Thus, we extend FoA using Weakly Supervised Learning (WSL) to generate salient region annotation only from shots annotated with the presence or absence of the target concept. Moreover, rather than the target concept, other concepts are more salient in several shots. Features of these shots falsely emphasise concepts other than the target. Hence, we develop a filtering method to eliminate shots where the target concept is unlikely to be salient. Experimental results show the effectiveness for each of our contributions, that is, SIN using FoA, FoA extended by WSL, and filtering.

Keywords—*Semantic indexing; Focus of attention; Weakly supervised learning; Filtering.*

I. INTRODUCTION

For effective processing of large-scale video data, one key technology is *Semantic Indexing* (SIN) to detect human-perceivable concepts in shots [1], [2]. Concepts are textual descriptions of semantic meanings that can be perceived by humans, such as *Person*, *Car*, *Building* and *Explosion_Fire*. Below, concept names are written in italics to distinguish them from the other terms. Many sources reported that the state-of-the-art video processing can be achieved using concept detection results as an intermediate representation of a shot [3]. Regarding this, traditional features just represent visual characteristics that significantly vary depending on various changing factors like camera techniques and shooting environments. On the other hand, the intermediate representation describes the presence of semantically meaningful concepts. Thus, if we could obtain accurate results where concepts are detected irrespective of changing factors, those results would facilitate categorising/retrieving shots that are visually dissimilar, but show similar semantic meanings. Motivated by this, much research effort has been made on SIN [1], [3], [4], [5], [6], [7].

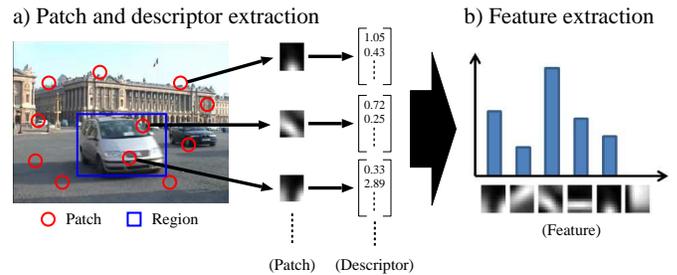


Figure 1. An illustration of feature extraction based on descriptors extracted from patches.

SIN is formulated as a binary classification problem where shots displaying a target concept are distinguished from the rest of the shots. One of the most important issues is feature extraction. Using Figure 1, we present an overview of the currently most popular approach [3], [4], [5], [6] while defining necessary terms for the following discussion. The approach consists of two main steps below:

1. Patch and descriptor extraction: This step aims to collect visual characteristics of *patches* that are small regions in a shot like red circles in Figure 1 (a). The rationale behind this is that as long as many patches are collected, some of them should keep their visual characteristics similar, irrespective of changing factors. From each patch, a *descriptor* is extracted as a vector, which numerically represents its visual characteristic. This is exemplified in Figure 1 (a), where three of patches are enlarged and their descriptors are shown on the right. It should be noted that compared to patches, we use the term ‘region’ to indicate a much larger region like the one of the car surrounded by the blue rectangle in Figure 1 (a).

2. Feature extraction: This step aggregates descriptors extracted from various patches to form a feature, which represents the distribution of those descriptors. For example, the histogram-type feature in Figure 1 (b) reveals that many descriptors characterise patches similar to the third one from the left, and there is no patch that is similar to the rightmost one in terms of descriptors. This kind of feature is effective for capturing detailed parts of a target concept. Especially, even if the target concept is partially invisible due to the occlusion by other concepts or the camera setting, the feature includes descriptors extracted from patches corresponding to the visible part of the target.

However, many concepts other than the target are displayed in a shot. For example, the shot in Figure 1 (a) includes the



Figure 2. Example shots where *Car* is shown in non-salient regions.

target concept *Car* and many others like *Building*, *Road* and *Sky*. Nonetheless, most of the existing SIN methods [3], [4], [5], [6] do not consider whether each patch belongs to the target concept or not. As a result, the feature is affected by patches of other concepts, and the detection performance of the target concept is degraded.

We aim to develop a SIN method that effectively spotlights a target concept while suppressing effects of the other irrelevant concepts. To this end, we incorporate *Focus of Attention* (FoA) (also called *visual attention*) into SIN. FoA implements ‘selective attention’ that is a brain mechanism to determine which regions in a shot (or video frame) are of most interest [8], [9], [10]. It is said that eyes are receiving visual data with the size 10^8 - 10^9 bits every second [9]. It is impossible for a human to completely analyse this huge size of data. Nevertheless, the human can effortlessly recognise meanings in a shot by fixating (or directing his/her gaze to) important regions. We apply this brain mechanism to SIN with the following logic: The fact that the human perceives the appearance of a target concept in the shot means that he/she fixates its region. Based on this, FoA is used to increase priorities of such regions and decrease those of the other regions, in order to construct a feature that emphasises the appearance of the target concept. In what follows, regions that attract fixations are called *salient regions*.

It should be noted that we focus only on appearances of a target concept in salient regions. In other words, we do not address its appearances in non-salient regions. For example, assuming that *Car* is the target concept, two shots in Figure 2 display it only in small background regions surrounded by red rectangles. These regions are clearly non-salient. Humans do not pay attention to or are not aware of the target concept appearing in such non-salient regions. Hence, these appearances are considered as meaningless and useless for subsequent processes like video categorisation, browsing and retrieval.

FoA consists of two main processes, *bottom-up* and *top-down*. The former implements human attention driven by stimuli acquired from the external environment. Since these stimuli can be thought as the visual information that eyes receive from a shot, they are equated with features extracted from the shot. However, salient regions detected based only on features are not so accurate because of the *semantic gap*, which is the lack of agreement between automatically extractable features and human-perceived semantics [11]. Thus, the top-down process implements attention driven by prior knowledge and expectation in the internal human mind. This biases the selection of salient regions based on human’s intention, goal and situation. In our case, the top-down process utilises the knowledge about spatial relations between a target concept and surrounding ones in order to selectively localise salient regions most likely to contain the target. Finally, salient

regions obtained by the bottom-up and top-down processes are combined to model their interaction.

To incorporate FoA into SIN, we address the following two issues: The first issue is the data availability of the top-down process. One typical formulation of this process is to adopt the machine learning framework, where salient regions in test shots are detected by referring to training shots in which salient regions are annotated in advance [12], [13], [14] or recorded by an eye tracker [14], [15]. However, a large number of training shots is needed to detect diverse kinds of salient regions. Due to a tremendous number of video frames in shots, it requires prohibitive cost to manually prepare many training shots. In addition, using an eye-tracker requires both labour and monetary costs. Thus, we develop an FoA method using *Weakly Supervised Learning* (WSL), where a classifier to predict precise labels is constructed only using loosely labelled training data [16]. In our case, this kind of training data are shots that are annotated only with the presence or absence of a target concept. These shots are used to build a classifier that can identify the region of the target concept in a shot, such as the blue rectangular region in Figure 1 (a) in the case where *Car* is the target. Regions identified by the classifier are used as annotated salient regions in the top-down process.

The second issue is the discrepancy that salient regions do not necessarily coincide with regions of a target concept. The reason is twofold: Firstly, it is difficult to objectively judge whether the target concept is salient or not. In other words, training shots can be annotated only with the presence of the target concept without considering its saliency. Consequently, like two shots in Figure 2, the target concept is shown in small background regions in several training shots. It is impossible or unreasonable to regard such regions as salient. The second reason for the discrepancy is possibly occurring errors in FoA. Even if the region of the target concept is salient for humans, another region may be falsely regarded as salient. A feature based on such a salient region incorrectly emphasises a non-target concept. To alleviate this, we develop a method that filters out shots where the target concept is unlikely to appear in salient regions, using regions predicted by the classifier in WSL. This enables us to appropriately capture characteristics of the target concept.

This paper is an extended version of our previous paper that only briefly illustrates our SIN method based on FoA due to the space limitation [1]. Specifically, the survey of related methods was quite insufficient in [1]. In contrast, the next section of this paper gives a comprehensive comparison of our method to various methods in four research fields, namely FoA, salient object detection, discriminative saliency detection, and SIN. In addition, while only a brief and conceptual explanation of our method was introduced in [1], its details and mathematical formulations are presented in Section III of this paper. Furthermore, although the experimental results in Section IV is the same to those in [1], Section V offers new ideas of how our method can be applied to different state-of-the-art features. Finally, for the sake of clarity, the Appendix provides a list of many abbreviations used in this paper.

II. RELATED WORK

FoA has been studied in the fields of computer vision, psychology and neurobiology for a long time. In particular, the development of a principled top-down process is one of the most important research topics [9]. Below, some types of

knowledge used in the top-down process of existing methods are presented. First, *contextual cueing* means that a user can easily search a particular object, if he/she saw the same or similar spatial layout of objects in the past [12], [13], [14], [15]. Salient regions in a test shot are adaptively extracted based on salient regions in training shots with similar spatial layouts. The *symmetry* indicates that while viewing a symmetric object, eye fixations are concentrated on the centre of symmetry [17]. Based on this, salient regions are preferentially located around centres of regions, which individually have a symmetric pattern of intensity or colour values. In addition, the *focusness prior* represents that a camera is often focused on the most salient object [18]. According to this, regions with low degrees of blur are more likely to be regarded as salient. Furthermore, the *centre prior* expresses that the main content is displayed near the centre, and is used to emphasise regions around the centre as salient [19]. Please refer to [9], [10] for other types of knowledge in the top-down process.

Among the knowledge described above, we use contextual cueing because it can be generally applied to any kind of videos. In particular, we target ‘unconstrained’ web videos that can be taken by arbitrary camera techniques and in arbitrary shooting environments [20]. Apart from contextual cueing, the symmetry highly depends on directions of a concept. Although the frontal appearance of the concept is symmetric, its side appearance may not be so. In addition, the focusness and centre priors are considered as valid only for professional videos, which follow shooting and editing rules to clearly convey the content to viewers. On the other hand, web videos are usually created by amateurs without taking such rules in account. As a result, the main content in a shot is often captured unfocused, and is not necessarily displayed near the centre of a video frame. In contrast, the generality of contextual cueing can be enhanced using a large amount of training shots, so that a variety of salient regions in web videos can be covered. Also, while existing methods based on contextual cueing require training shots that are annotated with salient regions [12], [13], [14] or eye fixations [14], [15], we use WSL to generate such annotation from shots labelled only with the presence or absence of a concept.

Our method is now compared to two extensions of FoA. The first is *Salient Object Detection* (SOD) that extracts the region of an object attracting the most user attention [10], [21], [22]. Since the principle of FoA is to detect regions where people look as salient, it is not guaranteed that salient regions correspond to semantically meaningful objects. It often happens that salient regions only characterise parts of an object, where these parts are visually distinctive or contrastive compared to the surrounding ones. Thus, SOD detects regions that not only are salient but also characterise meaningful objects. Also, there is an experimental evidence indicating that salient regions are strongly correlated with attractive objects [23]. Our method differs from SOD in the following two points: First, although most SOD methods need training shots where regions of salient objects are annotated [21], [22], our method using WSL only needs training shots annotated with the presence or absence of a target concept. Second, SOD just detects the region of a salient object without identifying its category. In contrast, the category of a target concept is considered in our method based on WSL. Here, regions of the target concept are identified as the ones that are commonly contained in training shots annotated with its presence, but

are not contained in training shots annotated with its absence. Using these identified regions, depending on the target concept, we adaptively find regions that are not only salient but also likely to contain it.

The second extension of FoA is *Discriminant Saliency Detection* (DSD) that extracts salient regions based on the discrimination power of descriptors for recognising a target concept (object) [24], [25]. Roughly speaking, DSD first regards the extraction of descriptors from patches as the bottom-up process, because they can be directly derived from images/videos (i.e., stimuli from the external environment). Then, the top-down process is performed as the selection of ‘salient’ descriptors, which best discriminate between the target concept and the others. Salient regions are computed based on locations of salient descriptors. However, as seen from the above-mentioned overview, DSD is significantly biased towards the recognition task, and does not care whether the region of the target concept is perceptually salient or not. In other words, DSD regards the target concept as salient even if it is shown in a small background region. Compared to this, we develop a filtering method that eliminates shots where the target concept is unlikely to appear in salient regions, by checking the coincidence between salient regions detected by FoA and regions identified by WSL. Appearances of the target concept in non-salient regions are considered as useless, because they do not attract user attention.

Finally, SIN is established in TRECVID that is a NIST-sponsored annual worldwide competition on video analysis and retrieval [2]. New SIN methods are being developed every year. The most popular approach is to extract a feature of a shot by encoding the distribution of descriptors using a histogram [4], using Gaussian Mixture Model (GMM) representing both means and variances of the descriptor distribution [5], [6], and using Fisher vector considering the first and second order differences between the distribution and the reference one [7]. Recently, researchers have started to adopt deep learning where a multi-layer convolutional neural network is used to extract a feature hierarchy with higher-level features formed by the composition of lower-level ones [7]. Despite this advancement of features, to our best knowledge, no method utilises FoA to enhance the quality of features. In this paper, we demonstrate the effectiveness of FoA to improve the most standard histogram-type feature.

III. SIN BASED ON FOA EXTENDED BY WSL

Figure 3 presents an overview of our SIN method. We call training shots annotated with the presence and absence of a target concept *positive shots* and *negative shots*, respectively. Since the target concept is *Car* in Figure 3, it is displayed and not displayed in positive and negative shots, respectively. For each of these training shots, FoA is performed to obtain its *saliency map* as shown in the middle of Figure 3. This map is an image representing the degree of saliency at each pixel. The higher saliencies of pixels are, the brighter they are depicted. Figure 3 shows saliency maps obtained for the positive and negative shots presented at the left. In the positive shot, the region of the moving car is regarded as salient. In the negative shot where a person is making a hand gesture, the region of his moving hand is regarded as salient.

After FoA, the feature of a training shot is extracted by weighting each descriptor based on the saliency of the patch from which the descriptor is derived. More concretely, red dot-

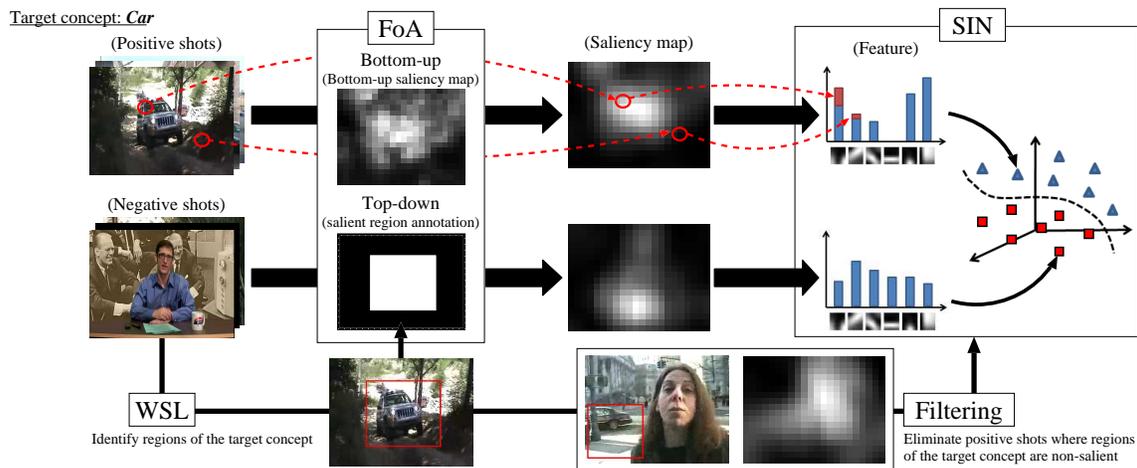


Figure 3. An overview of our SIN method using FoA extended by WSL

ted arrows starting from the positive shot in Figure 3 illustrate that the descriptor extracted from a patch in a salient region and the one extracted from a patch in a non-salient region have large and small influences on the feature, respectively. Finally, SIN is carried out by regarding training shots represented with such features as points in the multi-dimensional space. For the sake of visualisation, Figure 3 only depicts a three-dimensional space where triangles and rectangles indicate positive and negative shots, respectively. As shown in the dotted curve in this space, a *detector* is constructed to discriminate between positive and negative shots, and used to examine the presence of the target concept in test shots.

In Figure 3, the FoA module first conducts the bottom-up process on each training shot to compute its ‘intermediate’ saliency map, called *bottom-up saliency map*. This map is based only on features because the bottom-up process implements how eyes react to the visual information (see Section I). Specifically, regions that are visually different from surrounding ones are regarded as salient. However, it is difficult to accurately detect salient regions only using features. For example, Figure 3 shows the bottom-up saliency map for the positive shot, where in addition to the region of the car many background regions are also regarded as salient. Hence, the top-down process is needed to refine bottom-up saliency maps. To this end, WSL is firstly applied to training shots in order to prepare salient region annotation necessary for the top-down process. As a result, a classifier that identifies regions of the target concept is built. In Figure 3, the image under the FoA module and the black-and-white image over it indicate that, regions identified in positive shots (i.e., red rectangle) are used as annotated salient regions. Based on this, the top-down process is performed to refine a bottom-up saliency map into the final one.

The SIN module in Figure 3 involves filtering. Let us consider the positive shot and its saliency map on the left of the ‘Filtering’ box in Figure 3. The positive shot shows *Car* only in the small background region depicted by the red rectangle. Correspondingly, this region is not so salient while the region of the woman in the foreground is regarded as the most salient. The feature extracted from this positive shot falsely emphasises the non-target concept *Person*, and misleads a detector to detect it. Thus, filtering is performed to eliminate positive shots where the target concept is unlikely to appear

in salient regions. Below, we describe the bottom-up and top-down processes, WSL method, and SIN method with filtering.

A. Bottom-up Process

Figure 4 illustrates an overview of the bottom-up process where the positive shot on the left of Figure 3 is used as an example. We use a retina model to design how the bottom-up saliency map of a shot is created based on the visual information received by human eyes [14]. As shown in the upper part of Figure 4, it is known that the visual information is sequentially processed by horizontal, bipolar and Amacrine cells in the retina. The first cells perform smoothing to emphasise contrasts in the visual information, the second cells detect edges (or contours), and the last ones conduct the second smoothing to emphasise detected edges. Finally, according the feature integration theory [26], the above cells process different types of visual information in parallel, and the brain integrates processing results to focus the attention on certain regions. In what follows, we explain how to implement each cell’s mechanism and how to integrate processing results.

First of all, as the encoding of the visual information that arrives at eyes, the following six features related to cell responses in the retina are extracted [14]:

$$\text{Intensity: } I = \frac{r + g + b}{3}, \quad (1)$$

$$\text{Red-Green (RG) contrast: } RG = \frac{r - g}{\max(r, g, b)}, \quad (2)$$

$$\text{Blue-Yellow (BY) contrast: } BY = \frac{b - \min(r, g)}{\max(r, g, b)}, \quad (3)$$

$$\text{Flicker: } F = I - I', \quad (4)$$

$$\text{Motion direction: } \Theta = \tan^{-1}\left(\frac{v}{u}\right), \quad (5)$$

$$\text{Motion strength: } \Gamma = \sqrt{u^2 + v^2}, \quad (6)$$

where r , g and b represent the red-, green- and blue-channel values of a pixel in a video frame, respectively. In Equation (4), I' is the intensity in the previous video frame. In Equations (5) and (6), u and v are the horizontal and vertical displacements of the optical flow starting at a pixel. It should be noted that the above-mentioned features are extracted from each pixel in the video frame. Thus, each feature is represented as an

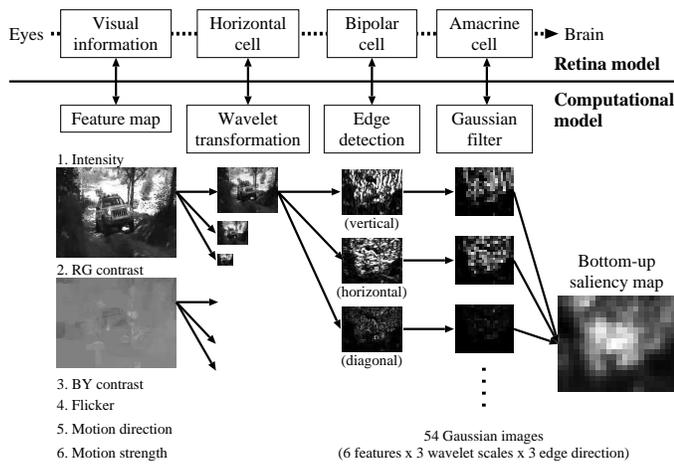


Figure 4. An overview of the bottom-up process in our FoA method

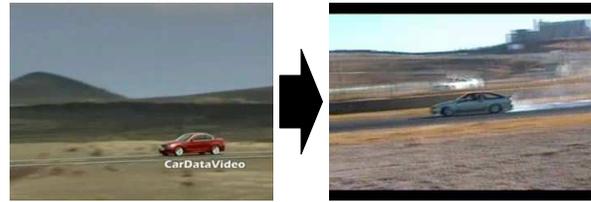
image, called ‘feature map’, which has the same size to the video frame, as shown at the left of Figure 4.

Then, smoothing by horizontal cells is implemented as wavelet transform on each feature map. As shown in Figure 4, the feature map is scaled down into 1/2, 1/4 and 1/8 sizes of images, termed as ‘wavelet images’. This down-scaling is useful for emphasising contrasts in the feature map while removing noises. In addition, wavelet images with three scales facilitate flexibly detecting salient regions with different sizes. Subsequently, edge detection by bipolar cells is simulated by applying high-pass filtering to each wavelet image. Three Sobel filters are used to extract edges (high-frequency components) in the vertical, horizontal and diagonal directions, as seen from Figure 4. This converts the wavelet image into three ‘edge images’ where an edge represents the saliency of the corresponding pixel, because this edge indicates the difference between the pixel and surrounding ones. Afterwards, to highlight such edges and suppress noises, the second smoothing by Amacrine cells is conducted using Gaussian filter for each edge image. We name the resulting image as a ‘Gaussian image’.

As a result of the aforementioned steps, 54 Gaussian images are obtained for each video frame (i.e., 6 feature maps \times 3 wavelet scales \times 3 edge directions). These Gaussian images are now integrated into a bottom-up saliency map. For computational efficiency, each Gaussian image is firstly scaled to the size 22×18 pixels. Note that each pixel in this scaled image corresponds to a region in the original video frame, based on the relative positional relation between the pixel and the region. In this sense, pixels in the scaled Gaussian image are called *macro-blocks*. The subsequent bottom-up saliency map extraction and top-down process use macro-blocks as the unit. Also, keep in mind that in Figures 3, 4 and 7, each saliency map with the size 22×18 pixels is resized to the original video frame size. As is clear from red dotted arrows in Figure 3, this resizing allows us to determine the saliency of each patch in the original video frame. The bottom-up saliency map of the video frame is created by taking the average of 54 Gaussian images for each macro-block. In addition, this map is normalised so that the most and least salient macro-blocks have 0 and 1, respectively.

Target concept: *Car*

Task 1: Long shots for cars moving from right to left in outdoor situations
 → Regions of moving cars are salient



Task 2: Close-up shots for car fronts in outdoor situations
 → Regions of cars are salient



Figure 5. Two conceptual examples of tasks

B. Top-down Process

The top-down process implements attention related to *tasks*. According to the contextual cueing described in Section II, we define a task as the expectation that, for shots with a certain type of spatial layouts, a human supposes to locate salient regions of a target concept at similar places. In Figure 5, where the target concept is *Car*, let us consider the situation where the human already saw the top-left shot and confirmed that the region of the moving car is salient. Based on this experience, the human expects that the region of the moving car in the top-right shot is also salient, because it has the similar spatial layout to the top-left shot. Similarly, when the human knows that the region of the car front in the bottom-left shot is salient, he/she should apply the same logic to the bottom-right shot. Like this, a task is the human’s expectation for salient regions of the target concept based on the similarity in camera techniques and shooting environments. However, only using such tasks lacks the examination of whether detected regions are visually (perceptually) salient or not. To resolve this, it is important to integrate the top-down and bottom-up processes. Hence, the top-down process in our method works to refine the bottom-up saliency map, so that salient regions detected by the bottom-up process are biased based on task-related attention described above.

First, we explain how to model task-related attention, which generally occurs by adjusting responses of cells in the retina to a specific type of stimuli [13], [14], [15]. Based on this, we re-use the retina model in Figure 4 and model task-related attention as the adjustment of 54 Gaussian images to a target concept [14]. Let us assume P positive shots, where each of them is associated with L ($= 54$) Gaussian images that are individually represented with N ($= 22 \times 18$) macro-blocks. For the i th positive shot ($1 \leq i \leq P$), we create an L -dimensional vector $\mathbf{x}_{in} = (x_{in}^1, \dots, x_{in}^L)$ by aggregating values at the n th macro-block ($1 \leq n \leq N$) in L Gaussian images. For example, assuming that the positive shot in Figure 4 is the i th one, \mathbf{x}_{i1} is the collection of values at the top-left macro-block in 54 Gaussian images. Note that the exact definition is $\mathbf{x}_{i'n}$ corresponding to the n th macro-block for the i' th video frame in the i th positive shot. But, this makes

the discussion unnecessarily complex. Thus, we use x_{in} for simplicity. Extending x_{in} to $x_{i'n}$ is straightforward, and our experiments are conducted using video frames sampled at every second.

A task is modelled as a function f_t to adjust x_{in} . Here, f_t is a linear function $f_t(x_{in}) = \sum_{l=1}^L w_t^l x_{in}^l$ where $\{w_t^l\}_{l=1}^L$ is a parameter set estimated using salient region annotation obtained by WSL in the next section. However, it is difficult to deterministically decide which task is used for a positive shot. In other words, it is impossible to objectively find to what extent each task is applicable for the positive shot, in terms of differences in appearances of the target concept, camera techniques and shooting environments. For example, in comparison to the top-left shot in Figure 5, let us consider a shot where a car is moving from left to right, the camera is placed closer to the car, and the situation is urban. It is unknown whether ‘‘Task 1’’ in Figure 5 can be used for this shot. Hence, we adopt a ‘soft assignment’ approach where functions $\{f_t\}_{t=1}^T$ for T tasks are probabilistically related to each positive shot. That is, x_{in} is adjusted by $\sum_{t=1}^T c_{it} f_t(x_{in})$ where c_{it} represents the weight of f_t for the i th positive shot.

Using task-related attention based on f_t s, we explain how to refine a bottom-up saliency map. Let b_{in} be the value at the n th macro-block in the bottom-up saliency map for the i th positive shot. We carry out the refinement of b_{in} as the weighted combination of b_{in} and the adjustment of x_{in} , that is, $\sum_{t=1}^T c_{it} f_t(x_{in}) + \alpha_{ib} b_{in}$. Here, α_{ib} is the weight representing the importance of the bottom-up saliency map. The top-down process estimates the following two components: The one is a set of parameter sets for T functions $F = \{\{w_t^l\}_{l=1}^L\}_{t=1}^T$, and the other is a set of weight vectors $C = \{c_i = (c_{i1}, \dots, c_{iT}, \alpha_{ib})\}_{i=1}^P$ where c_i represents weights of functions and the bottom-up saliency map for the i th positive shot. These F and C are estimated so as to accurately approximate salient region annotation $y_{in} \in \{0, 1\}$, where $y_{in} = 1$ means that the n th macro-block in the i th positive shot is salient, otherwise non-salient. Note that by regarding the binary value y_{in} as continuous, F and C are estimated as the regression problem of such continuous values using $\sum_{t=1}^T c_{it} f_t(x_{in}) + \alpha_{ib} b_{in}$.

In particular, for effective estimation of F , we employ *multi-task learning* that simultaneously estimates the parameter set $\{w_t^l\}_{l=1}^L$ for each of T functions by considering their correlation [14]. Compared to estimating such sets independently, the correlation can make it clearer what kind of salient regions are handled by each function. To sum up, the following optimisation is performed to estimate F and C [14]:

$$\min_{F, C} \frac{1}{PN} \sum_{i=1}^P \sum_{n=1}^N l \left(\sum_{t=1}^T c_{it} f_t(x_{in}) + \alpha_{ib} b_{in}, y_{in} \right), \quad (7)$$

where $l(\cdot)$ indicates the loss (error) computed as the squared difference between the refined saliency value ($\sum_{t=1}^T c_{it} f_t(x_{in}) + \alpha_{ib} b_{in}$) and the annotated one (y_{in}). Equation (7) aims to extract F and C that minimise the average refinement error for $P \times N$ macro-blocks. This optimisation can be solved by an EM-like algorithm, which iteratively switches between the estimation of C keeping F fixed, and the one of F keeping C fixed (see [14] for more details).

The bottom-up saliency map of each test shot is refined using the estimated F and C . Let us assume the j th test shot

where the n th macro-block is characterised by x_{jn} based on 54 Gaussian images and b_{jn} of the bottom-up saliency map. Based on the contextual cueing in Section II, the same refinement mechanism is used for shots with similar spatial layouts. Thus, we first find the \hat{i} th positive shot that has the most similar spatial layout to the j th test shot. Then, the saliency value of the n th macro-block is refined into s_{jn} using F and the weight vector $c_{\hat{i}}$ for the \hat{i} th positive shot [14]:

$$s_{jn} = \sum_{t=1}^T c_{\hat{i}t} f_t(x_{jn}) + \alpha_{\hat{i}b} b_{jn}. \quad (8)$$

The computation of similarities regarding spatial layouts requires to consider the global visual characteristic of a shot. To this end, for each of six feature maps in Figure 4, a histogram is created by quantising the value of every pixel into eight bins. This histogram represents the overall distribution of values in the feature map with respect to intensity, red-green contrast, blue-yellow contrast, or so forth. We use the concatenation of such histograms as the feature of the shot, and compute the similarity between two shots as their cosine similarity.

C. Weakly Supervised Learning

Motivated by the success of Support Vector Machines (SVMs) in object detection/recognition and SIN [20], [27], we employ the WSL method that is an extended SVM for WSL [16]. Usually, an SVM is trained using training shots associated with binary labels, that is, the presence or absence of a target concept. Then, it is used to predict the same type of binary labels for test shots. On the other hand, the method in [16] uses training shots with binary labels to build an SVM that can identify regions of the target concept. The main idea is that the method simultaneously localises the most distinctive regions and builds an SVM to distinguish those regions. More specifically, the SVM is trained so as to characterise regions that are contained in every positive shot, but are not contained in any negative shot. These regions are likely to contain the target concept.

First of all, we explain how regions of a target concept are localised by the method in [16]. Let x be an arbitrary shot without specifying it is positive or negative. We define the localisation as the problem to find the best ‘rectangular’ region \hat{r} from the set of all possible regions $\mathcal{R}(x)$ in x . With respect to this, one rectangular region is defined by four parameters, the top-left, top-right, bottom-left and bottom-right positions. Thus, simply speaking, $\mathcal{R}(x)$ includes $W^2 H^2$ rectangular regions if the frame size of x is $W \times H$ pixels. Since efficient search of \hat{r} will be discussed later, we here concentrate on the localisation mechanism. Assuming that a feature vector $\varphi(r)$ can be computed for any region $r \in \mathcal{R}(x)$ using descriptors in r , a linear SVM with the discrimination function $w\varphi(r) + b$ is used to examine whether r contains the target concept. Here, b is a bias term and w is a weight vector in which each dimension represents the relevance to the presence of the target concept. As $\varphi(r)$ has larger values on more relevant dimensions, the target concept is more likely to appear in r . Therefore, \hat{r} is determined as the region that maximises the discrimination function [16]:

$$\hat{r} = \operatorname{argmax}_{r \in \mathcal{R}(x)} (w\varphi(r) + b). \quad (9)$$

Based on this localisation mechanism, let x_i^+ and x_j^- be the i th positive and j th negative shots for a target concept, respectively. The parameters of the SVM (i.e., w and b) is estimated by solving the following optimisation problem [16]:

$$\min_{w,b} \left(\frac{1}{2} \|w\| + C \sum_i \alpha_i + C \sum_j \beta_j \right), \quad (10)$$

$$\text{s.t.} \quad \max_{r \in \mathcal{R}(x_i^+)} (w\varphi(r) + b) \geq +1 - \alpha_i \quad (\alpha_i \geq 0), \quad (11)$$

$$\max_{r \in \mathcal{R}(x_j^-)} (w\varphi(r) + b) \leq -1 + \beta_j \quad (\beta_j \geq 0), \quad (12)$$

where α_i (or β_j) is a slack variable representing the degree of mis-classification for the region in the x_i^+ (or x_j^-). In addition, C is a parameter to control the effect of mis-classification. The optimal w and b yields the situation where at least one region in x_i^+ is classified as positive (Equation (11)), while all regions in x_j^- are classified as negative (Equation (12)). The optimisation is solved using a coordinate descent approach, which iterates examining each training shot to find the best region that maximises the current discrimination function, and updating this function using newly found best regions [16].

For efficient optimisation, it is important to quickly find the best region for each training shot. To this end, we employ the region search method developed in [16], [28]. First, we use ‘Bag-of-Visual-Word’ (BoVW) representation to express the feature $\varphi(r)$ by quantising ‘Scale-Invariant Feature Transform’ (SIFT) descriptors in r . Each SIFT descriptor represents the edge shape in a patch, reasonably invariant to changes in scale, rotation, viewpoint and illumination [4]. As pre-processing, SIFT descriptors are extracted from patches, which have the radius of 10 pixels and are located at every sixth pixel in each training shot. Then, one million SIFT descriptors are randomly sampled and grouped into 1000 clusters, where each cluster centre is a ‘Visual Word’ (VW) representing a characteristic SIFT descriptor. Afterwards, by assigning each SIFT descriptor in r to the most similar VW, $\varphi(r) = (\varphi_1(r), \dots, \varphi_D(r))$ ($D = 1000$) is created where $\varphi_d(r)$ represents the frequency of the d th VW.

With the BoVW representation, the discrimination function of a linear SVM can be transformed as follows [28]:

$$w\varphi(r) + b = \sum_{d=1}^D w_d \varphi_d(r) + b = \sum_{n=1}^N w(\text{VW}_n) + b, \quad (13)$$

where N is the number of SIFT descriptors in r , and $w(\text{VW}_n)$ is the weight in $w = (w_1, \dots, w_D)$ corresponding to the VW associated with the n th SIFT descriptor. For example, $w(\text{VW}_n) = w_1$ if the n th SIFT descriptor is assigned to the first VW. Equation (13) means that the discrimination function can be computed by simply adding weights of VWs linked to SIFT descriptors in r . This enables us to estimate the ‘upper bound’ for a set of regions [28]. No region in this set takes the discrimination function value larger than the upper bound. Hence, the best region \hat{r} maximising the discrimination function can be efficiently found by discarding many sets of regions for which upper bounds are small.

Finally, \hat{r} that is detected in the i th positive shot x_i^+ using the optimised w and b , is used as the annotated salient region in the top-down process. Note that since the top-down process is based on 22×18 macro-blocks (pixels), the video frame in x_i^+ is resized to this size by preserving the relative spatial

relation between \hat{r} and the frame. Then, if the n th macro-block falls in \hat{r} , $y_{in} = 1$ otherwise $y_{in} = 0$.

D. Semantic Indexing with Filtering

As a result of FoA with WSL, saliency maps have been computed for all shots. As illustrated in Figure 3, our SIN method extracts the feature of a shot as an extended BoVW representation by weighting descriptors based on its saliency map. Let $\{\text{VW}_n\}_{n=1}^N$ be a set of VWs that are associated with N SIFT descriptors extracted from the whole of the shot. Also, let us denote by $\{p_n\}_{n=1}^N$ a set of centre positions of patches from which the N SIFT descriptors are extracted. That is, VW_n is the VW associated with the n th SIFT descriptor, which is extracted from the n th patch having the centre position p_n . Since the size of the saliency map is 22×18 pixels (macro-blocks), it is resized to the same size as the video frame of the shot. By checking this resized saliency map, we obtain $\{s_n\}_{n=1}^N$ where s_n represents the saliency of p_n . Then, an ‘weighted’ D -dimensional vector $\phi = (\phi_1, \dots, \phi_D)$ is created. Regarding this, in the normal BoVW representation, the value of the dimension corresponding to VW_n is incremented, so that the resulting feature represents the frequency of each VW. Different from this, in our extended BoVW representation, the value of the dimension is increased by s_n . Thereby, if VW_n is extracted from the patch in a salient region where the target concept probably appears, VW_n ’s effect is large, otherwise small (see red dotted arrows in Figure 3). Like this, the weighted vector ϕ emphasises the appearance of the target concept while suppressing effects of other concepts. Finally, using positive and negative shots represented by such ϕ s, a detector is constructed as a non-linear SVM with Radial Basis Function (RBF) kernel [29].

Before constructing the detector, filtering is performed to eliminate positive shots where the target concept appears in non-salient regions, because their weighted vectors falsely emphasise other concepts. To this end, we make a simple assumption that the target concept is salient if its region is large. Hence, positive shots are filtered out if regions detected by the WSL method are smaller than a threshold. Also, this filtering is executed when applying the detector to test shots. But, the purpose is to distinguish test shots where salient regions certainly include the target concept from the other ones. For the latter, we take into account FoA failures where falsely detected salient regions would cause weighted vectors undesirably emphasising non-target concepts. Thus, weighted vectors are extracted only from test shots where regions detected by the WSL method are larger than the threshold. For the other shots, non-weighted vectors are extracted based on the normal BoVW representation. Finally, the list of sorted test shots in terms of the detector’s outputs is returned as the SIN result .

IV. EXPERIMENTAL RESULTS

In this section, we firstly examine the effectiveness of our FoA method extended by WSL, and then evaluate the performance of SIN based on this extended FoA method.

A. Evaluation of FoA based on WSL

To examine the adequacy of incorporating WSL into FoA, we target three concepts *Person*, *Car* and *Explosion_Fire*. For each concept, we use 1000 positive shots and 5000 negative shots in TRECVID 2009 video data [2]. The performance is

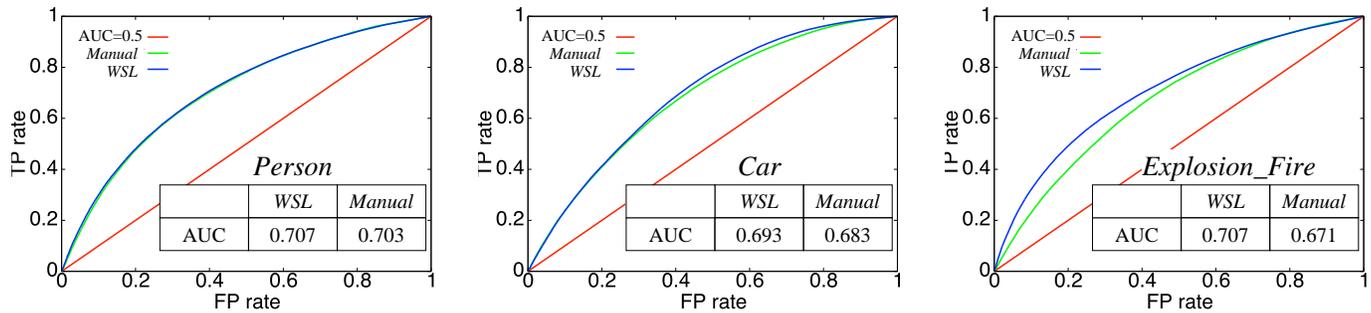


Figure 6. Performance comparison between WSL and Manual.

evaluated on 1000 test shots where the ground truth of salient regions is manually provided. We compare two FoA methods, *WSL* and *Manual*, which use positive shots where salient regions are annotated by WSL and by manual, respectively. Using manually annotated salient regions can be considered as the best approach. Hence, the comparison between *WSL* and *Manual* aims to investigate how useful salient regions obtained by WSL are, compared to those provided by the best manual approach.

Figure 6 shows Receiver Operating Characteristic (ROC) curves for *WSL* and *Manual*. Each curve is created by calculating True Positive (TP) and False Positive (FP) rates using different thresholds. Here, a macro-block in a saliency map is regarded as salient if its saliency is larger than a threshold. A TP is the number of macro-blocks that are correctly detected as salient, and an FP is the number of macro-blocks falsely detected as salient. A high performance is depicted by an ROC curve biased towards the top-left. In Figure 6, ROC curves of *WSL* and *Manual* are nearly the same for all concepts. As another evaluation measure, an Area Under Curve (AUC) represents the area under an ROC curve. A larger AUC indicates a superior performance where a high TP is achieved with a small FP. Figure 6 presents that *WSL*'s AUCs are nearly the same or even larger than those of *Manual*. The results described above verifies that salient regions annotated by WSL lead to the FoA performance that is comparable to the one based on the best manual approach.

It should be noted that several regions where a target concept does not appear are falsely detected by WSL, and used as annotated salient regions in the top-down process. For example, the red rectangular region in Figure 7 (a) is falsely regarded as showing a car. However, as seen from the bottom-up saliency map in Figure 7 (b), the saliency of this region is very low. More specifically, marco-blocks in this region have very small x_{jn} and b_{jn} in Equation (8). Thus, they cannot be regarded as salient even with the refinement by the top-down process, as shown in Figure 7 (c). Like this, errors in WSL are alleviated based on saliencies obtained by the bottom-up process. In other words, FoA works appropriately as long as regions obtained by WSL are mostly correct.

B. Evaluation of SIN using FoA

We evaluate the effectiveness of SIN utilising FoA using video data in TRECVID 2011 SIN light task [2]. According to the official guideline, 23 target concepts shown in Figure 8 are selected. For each target concept, a detector is constructed with 30000 training shots collected from 240918 shots in 11485 development videos. Here, positive shots are collected based on the result of web-based collaborative annotation

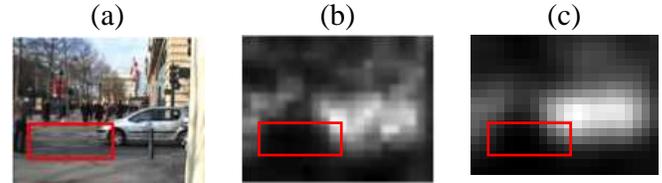


Figure 7. An example of alleviating errors in WSL based on bottom-up saliency maps.

effort where many users on the web collaboratively annotate shots in development videos [30]. Negative shots are collected as randomly selected shots in development videos. This is because the concept usually appears only in a small number of shots, so almost all of randomly selected shots can serve as negative [31]. Although annotation data collected by [30] contain negative shots, our preliminary experiment showed that they lead to worse performance than randomly selected shots. One main reason is the 'biased' shot selection based on active learning, where users are asked to only annotate shots similar to already collected positive shots [30]. In contrast, negative shots by 'non-biased' random selection yield more accurate concept detection. The constructed detector is tested on 125880 shots in 8215 test videos.

To examine the effectiveness of weighting descriptors based on FoA and that of filtering, we compare three methods *Baseline*, *Weight* and *Weight+Filter*. *Baseline* and *Weight* use features that are extracted as BoVW representations without and with weighting, respectively. *Weight+Filter* extends *Weight* by adding the filtering process. Figure 8 shows the performance comparison among *Baseline*, *Weight* and *Weight+Filter* in form of a bar graph. For each concept, the top, middle and bottom bars represent Average Precisions (APs) of *Baseline*, *Weight* and *Weight+Filter*, respectively. An AP approximates the area under a recall-precision curve. Regarding its computation, a SIN result for a target concept is a list of 2000 test shots ranked in terms of the detector's outputs. The AP is the average of precisions, each of which is computed at a position where a 'correct' test shot showing the target concept is ranked. A larger AP means a better SIN result where correct test shots are ranked at higher positions. Also, each of three bars at the bottom of Figure 8 presents the Mean of APs (MAP) over 23 concepts as an overall evaluation measure. Figure 8 indicates that *Weight* outperforms *Baseline* for many concepts. The MAP of the former (0.0708) is about 5% higher than that of the latter (0.0676). This validates the effectiveness of using FoA in SIN. In addition, *Weight+Filter*'s MAP (0.0731) exhibits that adding the filtering process improves *Weight*'s MAP by about 3%. This verifies the effectiveness of filtering.

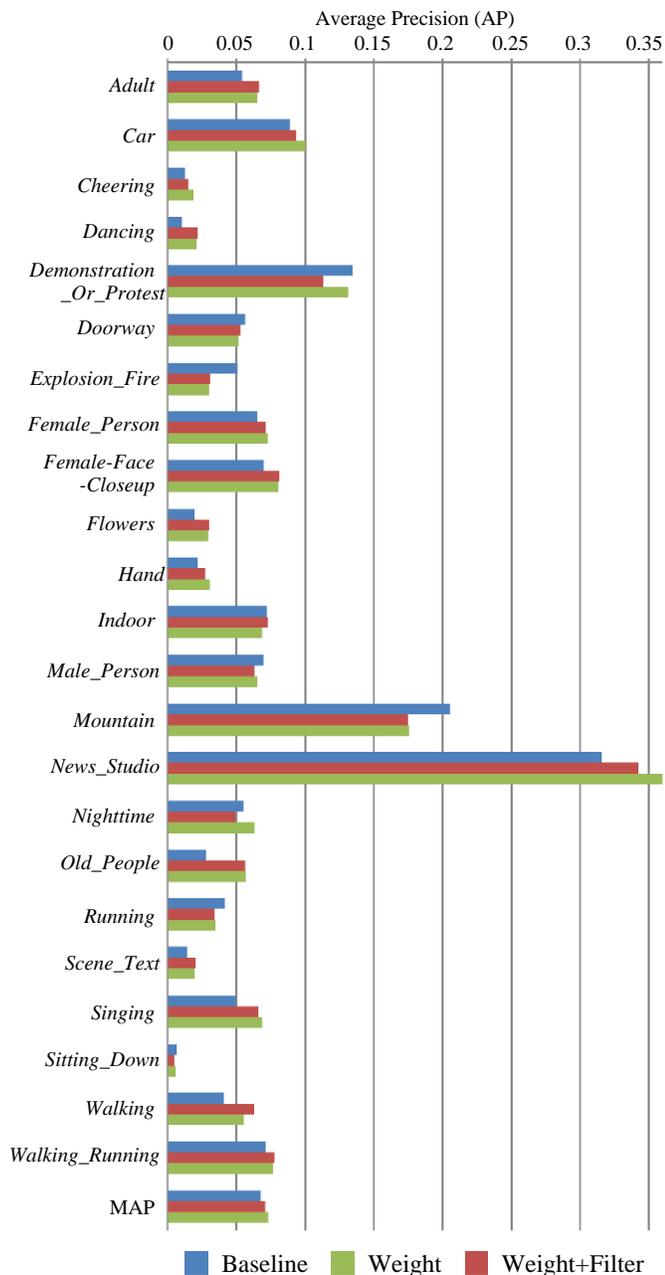


Figure 8. Performance comparison among *Baseline*, *Weight* and *Weight+Filter*.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a SIN method that detects a target concept based on FoA. Our method extracts the feature of a shot by weighting descriptors based on saliencies of patches, from which these descriptors are derived. This enables us to suppress adverse effects of regions irrelevant to the target concept, and emphasise its appearance. For effective integration of SIN and FoA, WSL is employed so that salient region annotation required for the top-down process can be generated from shots labelled only with the presence or absence of the target concept. In addition, filtering is conducted to eliminate shots where non-target concepts are emphasised, by examining the coincidence between salient regions detected by FoA and the target concept's regions identified by WSL. Experimental

results validated the effectiveness of all the three contributions, that is, using FoA in SIN, extending FoA with WSL, and filtering.

We will investigate the following two issues in the future: The first is that we used the most standard feature (i.e., BoVW representation) to justify the framework of using FoA in SIN. But, it is relatively straightforward to extend this framework to more sophisticated features, such as the ones based on GMM [5], [6], Fisher vector [7] and deep learning [7] described in Section II. Our ideas for this are summarised below. First, the extraction of GMM-based features starts with estimating a reference GMM using randomly sampled descriptors. Then, the GMM for a shot is computed by modifying the reference GMM based on descriptors extracted from the shot. FoA can be used to control the degree of modification based on the saliency of each descriptor, so that descriptors extracted from patches in salient regions have large influences on the resulting GMM. Second, the reference GMM is also used for Fisher vector-based features. Here, the feature of a shot is computed by averaging first (or second) order differences of descriptors to the mean of each Gaussian component in the reference GMM [32]. This averaging can be improved by considering the saliency of each descriptor. Last, one key factor in deep learning is how to define receptive fields, each of which represents a region that a neuron uses to extract a feature. FoA can be used to prioritise or select receptive fields of neurons by checking saliencies of regions. We will test each of the above-mentioned extensions.

The second issue is that FoA causes the performance degradation for some concepts such as *Explosion_Fire* and *Mountain* in Figure 8. One main region is non-rectangular shapes of these concepts, because our current WSL method can only identify rectangular regions. However, rectangular regions are too coarse to precisely localise non-rectangular concepts, and inevitably include other concepts. As a result, the top-down process does not work well. Hence, we will extend our WSL method by adopting an efficient search algorithm for regions with arbitrary shapes [33].

APPENDIX LIST OF ABBREVIATIONS

The list below shows abbreviations used in this paper. Each line presents an abbreviation, its full name, and the section where it appears for the first time.

SIN	: Semantic INDEXing (Section I)
FoA	: Focus of Attention (Section I)
WSL	: Weakly Supervised Learning (Section I)
SOD	: Salient Object Detection (Section II)
DSD	: Discriminant Saliency Detection (Section II)
GMM	: Gaussian Mixture Model (Section II)
SVM	: Support Vector Machines (Section III-C)
BoVW	: Bag-of-Visual-Word (Section III-C)
VW	: Visual Word (Section III-C)
SIFT	: Scale-Invariant Feature Transform (Section III-C)
RBF	: Radial Basis Function (Section III-D)
ROC	: Receiver Operating Characteristic (Section IV-A)
TP	: True Positive (Section IV-A)
FP	: False Positive (Section IV-A)
AUC	: Area Under Curve (Section IV-A)
AP	: Average Precisions (Section IV-B)
MAP	: Mean of Average Precision (Section IV-B)

REFERENCES

- [1] K. Shirahama, T. Matsumura, M. Grzegorzec, and K. Uehara, "Empowering semantic indexing with focus of attention," in Proc. of MMEDIA 2015, 2015, pp. 33–36.
- [2] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID," in Proc. of MIR 2006, 2006, pp. 321–330.
- [3] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Found. Trends Inf. Retr.*, vol. 2, no. 4, 2009, pp. 215–322.
- [4] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, 2010, pp. 1582–1596.
- [5] N. Inoue and K. Shinoda, "A fast and accurate video semantic-indexing system using fast MAP adaptation and gmm supervectors," *IEEE Trans. Multimed.*, vol. 14, no. 4, 2012, pp. 1196–1205.
- [6] K. Shirahama and K. Uehara, "Kobe university and muroran institute of technology at TRECVID 2012 semantic indexing task," in Proc. of TRECVID 2012, 2012.
- [7] C. G. M. Snoek et al., "Mediamill at TRECVID 2014: Searching concepts, instances and events in video," in Proc. of TRECVID 2014, 2014.
- [8] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," *ACM Trans. Appl. Percept.*, vol. 7, no. 1, 2010, pp. 6:1–6:39.
- [9] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, 2013, pp. 185–207.
- [10] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," 2014. [Online]. Available: <http://arxiv.org/abs/1411.5878>
- [11] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, 2000, pp. 1349–1380.
- [12] A. Oliva, A. Torralba, M. Castelano, and J. Henderson, "Top-down control of visual attention in object detection," in Proc. of ICIP 2003, 2003, pp. 253–256.
- [13] S. Chikkerur, T. Serre, C. Tan, and T. Poggio, "What and where: A bayesian inference theory of attention," *Vis. Res.*, vol. 50, no. 22, 2010, pp. 2233–2247.
- [14] J. Li, Y. Tian, T. Huang, and W. Gao, "Probabilistic multi-task learning for visual saliency estimation in video," *Int. J. Comput. Vis.*, vol. 90, no. 2, 2010, pp. 150–165.
- [15] W. Kienzle, M. O. Franz, B. Schölkopf, and F. A. Wichmann, "Center-surround patterns emerge as optimal predictors for human saccade targets," *J. Vis.*, vol. 9, no. 5, 2009, pp. 1–15.
- [16] M. Nguyen, L. Torresani, F. De la Torre, and C. Rother, "Weakly supervised discriminative localization and classification: a joint learning process," in Proc. of ICCV 2009, 2009, pp. 1925–1932.
- [17] G. Kootstra, A. Nederveen, and B. d. Boer, "Paying attention to symmetry," in Proc. of BMVC 2008, 2008, pp. 111.1–111.10.
- [18] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in Proc. of ICCV 2013, 2013, pp. 1976–1983.
- [19] H. Jiang, J. Wang, Z. Yuan, T. Liu, and N. Zheng, "Automatic salient object segmentation based on context and shape prior," in Proc. of BMVC 2011, 2011, pp. 110.1–110.12.
- [20] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimed. Inf. Retr.*, vol. 2, no. 2, 2013, pp. 73–101.
- [21] T. Liu et al., "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, 2011, pp. 353–367.
- [22] H. Jiang, H. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in Proc. of CVPR 2013, 2013, pp. 2083–2090.
- [23] L. Elazary and L. Itti, "Interesting objects are visually salient," *J. Vis.*, vol. 8, no. 3, 2008, pp. 1–15.
- [24] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, 2009, pp. 989–1005.
- [25] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in Proc. of CVPR 2012, 2012, pp. 3506–3513.
- [26] A. Treisman and G. Gelade, "Feature integration theory of attention," *Cogn. Psychol.*, vol. 12, no. 1, 1980, pp. 97–136.
- [27] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, 2010, pp. 1627–1645.
- [28] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in Proc. CVPR 2008, 2008, pp. 1–8.
- [29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, pp. 27:1–27:27.
- [30] S. Ayache and G. Quénot, "Video corpus annotation using active learning," in Proc. of ECIR 2008, 2008, pp. 187–198.
- [31] A. Natsev, M. R. Naphade, and J. Tešić, "Learning the semantics of multimedia queries and concepts from a small number of examples," in Proc. of MM 2005, 2005, pp. 598–607.
- [32] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in Proc. of BMVC 2011, 2011, pp. 76.1–76.12.
- [33] S. Vijayanarasimhan and K. Grauman, "Efficient region search for object detection," in Proc. of CVPR 2011, 2011, pp. 1401–1408.

A Novel Taxonomy of Deployment Patterns for Cloud-hosted Applications: A Case Study of Global Software Development (GSD) Tools and Processes

Laud Charles Ochei, Andrei Petrovski

School of Computing Science and Digital Media
Robert Gordon University
Aberdeen, United Kingdom

Emails: {l.c.ochei, a.petrovski}@rgu.ac.uk

Julian M. Bass

School of Computing, Science and Engineering
University of Salford
Manchester, United Kingdom

Email: J.Bass@salford.ac.uk

Abstract—Cloud patterns describe deployment and use of various cloud-hosted applications. There is little research that focuses on applying these patterns to cloud-hosted Global Software Development (GSD) tools. As a result, it is difficult to know the applicable deployment patterns, supporting technologies and trade-offs to consider for specific software development processes. This paper presents a taxonomy of deployment patterns for cloud-hosted applications. The taxonomy is composed of 24 sub-categories, which were systematically integrated and structured into 8 high-level categories. The taxonomy is applied to a selected set of software tools: JIRA, VersionOne, Hudson, Subversion and Bugzilla. The study confirms that most deployment patterns are related and cannot be fully implemented without being combined with others. The taxonomy revealed that (i) the functionality provided by most deployment patterns can often be accessed through an API or plugin integrated with the GSD tool, and (ii) RESTful web services and messaging are the dominant strategies used by GSD tools to maintain state and exchange information asynchronously, respectively. This paper also describes CLIP (CLoud-based Identification process for deployment Patterns), to guide software architects in selecting applicable cloud deployment patterns for GSD tools using the taxonomy and thereafter applies it to a motivating cloud deployment problem. Recommendations for guiding architects in selecting applicable deployment patterns for cloud deployment of GSD tools are also provided.

Keywords—Taxonomy; Deployment Pattern; Cloud-hosted Application; GSD Tool; Plugin; Continuous Integration

I. INTRODUCTION

Collaboration tools that support Global Software Development (GSD) processes are increasingly being deployed on the cloud [1][2][3]. The architectures/patterns used to deploy these tools to the cloud are of great importance to software architects, because they determine whether or not the system's required quality attributes (e.g., performance) will be exhibited [4][5][6].

Collections of cloud patterns exist for describing the cloud, and how to deploy and use various cloud offerings [7][8]. However, there is little or no research in applying these patterns to describe the cloud-specific properties of applications in the software engineering domain (e.g., collaboration tools for GSD, hereafter referred to as GSD tools) and the trade-offs

to consider during cloud deployment. This makes it very challenging to know the deployment patterns (together with the technologies) required for deploying GSD tools to the cloud to support specific software development processes (e.g., continuous integration (CI) of code files with Hudson).

Motivated by this problem, we propose a taxonomy of deployment patterns for cloud-hosted applications to help software architects in selecting applicable deployment patterns for deploying GSD tools to the cloud. The taxonomy will also help to reduce the time and risk associated with large-scale software development projects. We are inspired by the work of Fehling et al. [7], who catalogued a collection of patterns that will help architects to build and manage cloud applications. However, these patterns were not applied to any specific application domain, such as cloud-hosted GSD tools.

The research question this paper addresses is: **“How can we create and use a taxonomy for selecting applicable deployment patterns for cloud deployment of GSD tools?”** It is becoming a common practice for distributed enterprises to hire cloud deployment architects or “application deployers” to deploy and manage cloud-hosted GSD tools [9]. For example, the CI systems used by Salesforce.com (a major player in the cloud computing industry), runs 150000 + test in parallel across many servers and if it fails it automatically opens a bug report for software architects and developers responsible for that *checkin* [10].

We created and applied the taxonomy against a selected set of GSD tools derived from an empirical study [11] of geographically distributed enterprise software development projects. The overarching result of the study is that most deployment patterns are related and have to be combined with others during implementation, for example, to address hybrid deployment scenarios, which usually involves integrating processes and data in multiple clouds.

This article is an extension of the previous work by Ochei et al. [1]. The main contributions of this article are:

1. Creating a novel taxonomy of deployment patterns for cloud-hosted applications.
2. Demonstrating the practicality of the taxonomy by: (i)

applying it to position a set of GSD tools; and (ii) comparing the different cloud deployment requirements of GSD tools.

3. Describing CLIP, a novel approach for guiding architects in selecting applicable cloud deployment patterns for GSD tools using the taxonomy, and thereafter applying it to a motivating cloud deployment problem.

4. Presenting recommendations and best practice guidelines for identifying applicable deployment patterns together with the technologies for supporting cloud deployment of GSD tools.

The rest of the paper is organized as follows: Section II gives an overview of the basic concepts related to deployment patterns for Cloud-hosted GSD tools. In Section III, we discuss the research methodology including taxonomy development, tools selection, application and validation. Section IV presents the findings of the study focusing on positioning a set of GSD tools within the taxonomy. In Section V, we discuss the lessons learned from applying the taxonomy. The recommendations and limitations of the study are in Sections VI and VII, respectively. Section VIII reports the conclusion and future work.

II. DEPLOYMENT PATTERNS FOR CLOUD-HOSTED GSD TOOLS

A. Global Software Development

In recent times, Global Software Development has emerged as the dominant methodology used in developing software for geographically distributed enterprises. The number of large scale geographically distributed enterprise software development projects involving Governments and large multi-national companies is on the increase [12][13][14].

Definition 1: Global Software Development. GSD is defined by Lanubile [15] as the splitting of the development of the same software product or service among globally distributed sites. Global Software Development involves several partners or sites of a company working together to reach a common goal, often to make a product (in this case, software) [15, 16].

In geographically distributed enterprise software development, there are not only software developers, but many stakeholders such as database administrators, test analysts, project managers, etc. Therefore, there is a need to have software tools that support collaboration and integration among members of the team involved in the software development project. As long as a software project involves more than one person, there has to be some form of collaboration [17][16][11][18].

B. Cloud-hosted GSD Tools and Processes

Software tools used for Global Software Development projects are increasingly being moved to the cloud [3]. This is in response to the widespread adoption of Global Software Development practices and collaboration tools that support geographically distributed enterprises software projects [19]. This trend will continue because the cloud offers a flexible and scalable platform for hosting a broad range of software services including, APIs and development tools [2][3].

Definition 2: Cloud-hosted GSD Tool. “Cloud-hosted GSD Tool” refers to collaboration tools used to support GSD

processes in a cloud environment. We adopt the: (i) NIST Definition of Cloud Computing to define properties of cloud-hosted GSD tools; and (ii) ISO/IEC 12207 standard as a frame of reference for defining the scope of a GSD tool. Portillo et al. [20] identified three groups of GSD tools for supporting ISO/IEC 12207 processes:

(i) Tools to support Project Processes- These tools are used to support the management of the overall activities of the project. Examples of these processes include project planning, assessment and control of the various processes involved in the project. There are several GSD tools that fit into this group. For instance, JIRA and Bugzilla are software tools widely used in large software development projects issue and bug tracking. (ii) Tools to support Implementation Processes such as requirements analysis and integration process. For example, Hudson, is a widely used tool for continuously integrating different source code builds and components into a single unit. (iii) Tools for Support Processes - Software tools that fall into this group are used to support documentation management processes and configuration management processes involved in the software development project. For example, Subversion is a software tool used to track how the different versions of a software evolves over time.

These GSD tools, also referred to as Collaboration tools for GSD [20], are increasingly being deployed to the cloud for Global Software Development by large distributed enterprises. The work of Portillo et al. [20] presents the requirements and features of GSD tools and also categorizes various software tools used for collaboration and coordination in Global Software Development.

C. Architectures for Cloud-hosted Applications

Definition 3: Architectural Pattern. Architectural patterns are compositions of architectural elements that provide packaged strategies for solving recurring problems facing a system [5]. Architectural patterns can be broadly classified into 3 groups based on the nature of the architectural elements they use [5]: (i) module type patterns - which show how systems are organized as a set of codes or data units in the form of classes, layers, or divisions of functionality.

(ii) component-and-connector (C&C) type patterns - which show how the system is organized as a set of components (i.e., runtime elements used as units of computation, filters, services, clients, servers etc.) and connectors (e.g., communication channels such as protocols, shared messages, pipes, etc.).

(iii) allocation patterns - which show how software elements (typically processes associated with C&C and modules) relate to non-software elements (e.g., CPUs, file system, networks etc.) in its environment. In other words, this pattern shows how the software elements are allocated to elements in one or more external environment on which the software is executed.

Architectural and design patterns have long been used to provide known solutions to a number of common problems facing a distributed system [5][21]. The architecture of a system/application determines whether or not its required quality attributes (e.g., performance, availability and security) will be exhibited [4][5].

D. Cloud Deployment Patterns

In cloud computing environment, a cloud pattern represents a well-defined format for describing a suitable solution to a cloud-related problem. Several cloud problems exist such as how to: (i) select a suitable type of cloud for hosting applications; (ii) select an approach for delivering a cloud service; (iii) deploy a multitenant application that guarantees isolation of tenants. Cloud deployment architects use cloud patterns as a reference guide that documents best practices on how design, build and deploy applications to the cloud.

Definition 4: Cloud Deployment Pattern. We define a “Cloud deployment pattern” as a type of architectural pattern, which embodies decisions as to how elements of the cloud application will be assigned to the cloud environment where the application is executed.

Our definition of cloud deployment pattern is similar to the concept of design patterns [21], (architectural) deployment patterns [5], collaboration architectures [4], cloud computing patterns [7], cloud architecture patterns [22], and cloud design patterns [8]. These concepts serve the same purpose in the cloud (as in many other distributed environments). For example, the generic architectural patterns- client-server, peer-to-peer, and hybrid [5] - relate to the following: (i) the 3 main collaboration architectures, i.e., centralized, replicated and hybrid [4]; and (ii) cloud deployment patterns, i.e., 2-tier, content distribution network and hybrid data [7].

One of the key responsibilities of a cloud deployment architect is to allocate elements of the cloud-application to the hardware processing (e.g., processor, files systems) and communication elements (e.g., protocols, message queues) on the cloud environment, so that the required quality attributes can be achieved.

Figure 2 shows how the elements of Hudson (a typical of GSD tool) is mapped to the elements of the cloud environment. Hudson runs on an Amazon EC2 instance while the data it generates is regularly extracted and archived on a separate cloud storage (e.g., Amazon S3).

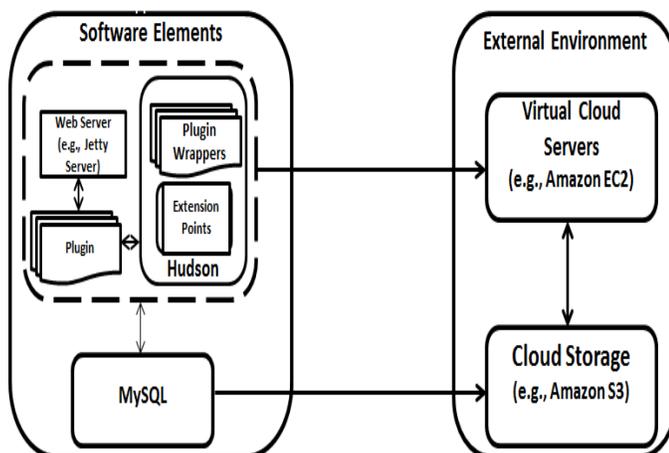


Fig. 1. Mapping elements of a GSD tool to External Environment

E. Taxonomy of Cloud Computing Patterns

What is a Taxonomy and its Purpose? The IEEE Software & Systems Engineering Standards Committee defines a Taxonomy as “a scheme that partitions a body of knowledge into taxonomic units and defines the relationship among these units. It aims for classifying and understanding the body of knowledge [23].” As understanding in the area of cloud patterns and cloud-hosted software tools for distributed enterprise software development evolves, important concepts and relationships between them emerge that warrant a structured representation of these concepts. Being able to communicate that knowledge provides the prospects to advance research [24].

Taxonomies and classifications facilitate systematic structuring of complex information. Taxonomies are mechanisms that can be used to structure, advance understanding and to communicate this knowledge [25]. According to Sjoberg [26], the development of taxonomies is crucial to documenting the theories that accumulate knowledge on software engineering. In software engineering, they are used for comparative studies involving tools and methods, for example, software evolution [27] and Global Software Engineering [28]. The work of Glass and Vessey [25] and Bourque and Dupuis [29] laid down the foundation for developing various taxonomies for software development methods and tools in software engineering. In this paper, we focus on using a taxonomy to structure cloud deployment patterns for cloud-hosted applications, in particular in the area of GSD tools.

Existing Taxonomies and Classifications of Deployment Patterns for Cloud-hosted Applications Several attempts have been made by researchers to create classifications of cloud patterns to build and deploy cloud-based applications. Wilder [22] describes eleven patterns: Horizontally Scaling Compute, Queue-Centric Workflow, Auto-Scaling, MapReduce, Database Sharding, Busy Signal, Node Failure, Colocate and Valet Key. The authors then illustrate how each pattern can be used to build cloud-native applications using the Page of Photos web application and Windows Azure. Each pattern is preceded by what the authors refer to as “primers” to provide a background of why the pattern is need. A description is provided about how each pattern is used to address specific architectural challenges that are likely to be encountered during cloud deployment.

A collection of over 75 patterns for building and managing a cloud-native application are provided by Fehling et al. [7]. The “known uses” of the implementation of each pattern is provided with examples of cloud providers offering products that exhibit the properties described in the pattern. This helps to further give a better understanding of the core properties of each pattern. We find the examples of known uses of patterns under “storage offering” category (e.g., blob storage, key-value storage) very useful in understanding how to modify a GSD tool in order to access a cloud storage. For example, Amazon S3 and Google cloud storage are products offered by Amazon and Google, respectively, for use as blob storage on the cloud. Blob storage is based on an object storage architecture, and so the GSD tool has to be modified to allow access using a REST API.

Homer et al. [8] describe: (i) twenty-four patterns that are

useful in developing cloud-hosted applications; (ii) two primers and eight guidance topics that provide basic information and good practice techniques for developing cloud-hosted applications; and (iii) ten sample applications that illustrate how to implement the design patterns using features of Windows Azure. The sample code (written in C#) for these sample applications is provided, thus making it easy for architects who intend to use similar cloud patterns to convert the codes to other web programming languages (e.g., Java, Python) for use in other cloud platforms.

Moyer [30] discusses a collection of patterns under the following categories: image (e.g., prepackaged images), architecture (e.g., adapters), data (e.g., queuing, iterator), and clustering (e.g., n-tier) and then use a simple Weblog application written using Amazon Web Services (AWS) with Python to illustrate the use of these patterns. For example, one of the architectural patterns- Adapters, is similar to “Provider Adapter” pattern described by Fehling et al [7], which can be used for interacting with external systems not provided by the cloud provider. The weblog application uses a custom cloud-centric framework created by the author called *Marajo*, instead of contributing extensions to existing Python frameworks (e.g., pylons). Apart from Marajo’s tight integration with AWS, it may be difficult for it to be widely used by software architects since it does not offer the rich ecosystem and large public appeal which other Python-based web frameworks currently offer.

Sawant and Shah discussed patterns for handling “Big Data” on the cloud [31]. These include patterns for big data ingestion, storage, access, discovery and visualization. For example, it describes how the “Federation Pattern” can be used to pull together data from multiple sources and then process the data. Doddavula et al. [32] present several cloud computing solution patterns for handling application and platform solutions. For instance, it discusses cloud deployment patterns for: (i) handling applications with highly variable workloads in public clouds; and (ii) handling workload spikes with cloud burst.

Erl et al. [33] present a catalogue of over 100 cloud design patterns for developing, maintaining and evolving cloud-hosted applications. The cloud patterns, which are divided into eight groups cover several aspects of cloud computing, such as scaling and elasticity, reliability and resiliency, data management, and network security and management. For example, patterns such as shared resources, workload distribution and dynamic scalability (which are listed under the “sharing, scaling and elasticity” category) are generally used for workload management and overall optimization of the cloud environment. The major strength of Erl et al.’s catalogue of cloud patterns is in its extensive coverage of techniques for handling the security challenges of cloud-hosted applications. It describes various strategies covering areas such as hypervisor attack vectors, threat mitigation and mobile device management. Other documentation of cloud deployment patterns can be found in [34][35][36][37][38][39].

Existing classifications of cloud patterns do not organize the individual patterns into a clean hierarchy or taxonomy. This is because most of the patterns tend to handle multiple architectural concerns [22]. This makes it difficult for an architect to decide whether the implementation of the cloud

can be done by modifying the cloud-application itself or the components of the cloud environment where the application is running.

Cloud patterns in existing classifications are applied to simple web-based applications (e.g., Weblog application [30]) without considering the different application processes they support. Moreover, these patterns have not been applied against a set of applications in software engineering domain, such as cloud-hosted GSD tools. GSD tools may have similar architectural structure but they (i) support different software development processes, and (ii) impose varying workloads on the cloud infrastructure, which would influence the choice of a deployment pattern. For example, Hudson being a compiler/build tool, would consume more memory than subversion when exposed to high intensive workload.

Motivated by these shortcomings, we extend the current research by developing a taxonomy of deployment patterns for cloud-hosted applications that reflects the two main components of an architectural deployment structure: the cloud-application and cloud environment. Thereafter, we apply the taxonomy to position a set of GSD tools.

III. METHODOLOGY

A. Development of a Taxonomy of Deployment Patterns for Cloud-hosted Applications

1) *Developing the Taxonomy:* We develop the taxonomy by using a modified form of the approach used by Lilien [40] in his work for building a taxonomy of specialized ad hoc networks and systems for a given target application class. The approach is summarized in the following steps:

Step 1: Select the target class of Software Tool- The target class is based on the ISO/IEC 12207 taxonomy for the software life cycle process (see Definition 3 for details). The following class of tools are excluded: (i) tools not deployed in a cloud environment (even if they are deployed on a dedicated server to perform the same function); and (ii) general collaboration tools and development environments (e.g., MS Word, Eclipse).

Step 2: Determine the requirements for the Taxonomy- The first requirement is that the taxonomy should incorporate features that restricts it to GSD tools and Cloud Computing. In this case, we adopt the ISO/IEC 12207 framework [20] and NIST cloud computing definition [41]. Secondly, it should capture the components of an (architectural) deployment structure [5] - software elements (i.e., GSD tool to be deployed) and external environment (i.e., cloud environment). Therefore, our proposed taxonomy is a combination of two taxonomies - Taxonomy A, which relates to the components of the cloud environment [41], and Taxonomy B, which relates to the components of the cloud application architecture [7].

Step 3: Determine and prioritize the set of all acceptable categories and sub-categories of the Taxonomy- We prioritized the categories of the taxonomy to reflect the structure of a cloud stack from physical infrastructure to the software

process of the deployed GSD tool. The categories and sub-categories of the two taxonomies are described as follows:

(1) *Application Process*: the sub-categories (i.e., project processes, implementation processes and support processes) represent patterns for handling the workload imposed on the cloud infrastructure by the ISO/IEC 12207 software processes supported by GSD tools [20]. For example, the unpredictable workload pattern described by Fehling et al. [7] can be used to handle random and sudden increase in the workload of an application or consumption rate the IT resources.

(2) *Core cloud properties*: the sub-categories (i.e., rapid elasticity, resource pooling and measured service) contain patterns used to mitigate the core cloud computing properties of the GSD tools [7].

(3) *Service Model*: the sub-categories reflect cloud service models- SaaS, PaaS, IaaS [41].

(4) *Deployment Model*: the sub-categories reflect cloud deployment models- private, community, public and hybrid [41].

(5) *Application Architecture*: the sub-categories represent the architectural components that support a cloud-application such as application components (e.g., presentation, processing, and data access), multitenancy, and integration. The multitenancy patterns are used to deploy a multitenant application to the cloud in such a way that guarantees varying degrees of isolation of the users. The three patterns that reflect these degrees of isolation are shared component, tenantisolated component and dedicated component [7].

(6) *Cloud Offerings*: the sub-categories reflect the major infrastructure cloud offerings that can be accessed- cloud environment, processing, storage and communication offering [7]. Patterns that fall under “communication patterns” are probably the best documented in this group. Examples include Priority Queue [8], Queue-Centric workflow, message-oriented middleware, which are used to ensure the reliability of messages exchanged between users.

(7) *Cloud Management*: contains patterns used to manage both the components and processes/runtime challenges) of GSD tools. The 2 sub-categories are - management components, which are used for managing hardware components (e.g., servers) and management processes, which are used for managing processes (e.g., database transactions) [7]. The node failure pattern described by Wilder [22] can be used to handle sudden hardware failures. The “Health Endpoint Monitoring” pattern [8] and the “resiliency management” pattern can be used to handle runtime failures or unexpected software failures.

(8) *Composite Cloud*: contains compound patterns (i.e., patterns that can be formed by combining other patterns or can be decomposed into separate components). The sub-categories are: decomposition style and hybrid cloud application [7]. The patterns under the decomposition style describe how the software and hardware elements of the cloud environment are composed (or can be decomposed) into separate components. A well-known example is the two-tier (or client/server) pattern, in which each component or process on the cloud environment is either a client or a server. Another example is the multisite deployment pattern [22], where users

form clusters around multiple data centres or are located in globally distributed sites. Hybrid cloud application patterns are integrations of other patterns and environments. For example, the “hybrid development environment” pattern can be used to integrate various clouds patterns to handle different stages of software development- compilation, testing, and production.

Step 4: Determine the space of the Taxonomy- The selected categories and their associated sub-categories define the space of the taxonomy. The taxonomy (Table I) is composed of 24 sub-categories, which were systematically integrated and structured into 8 high-level categories. The information that the taxonomy conveys has been arranged into four columns: deployment components, main categories, sub-categories, and related patterns.

2) *Description of the Taxonomy of Deployment Patterns for Cloud-hosted Applications*: Table I shows the taxonomy captured in one piece. In the following, we describe the key sections of the taxonomy.

Deployment Components of the Taxonomy: There are two sections of the taxonomy: the upper-half represents Taxonomy A, which is based on NIST Cloud Computing Definition, while the lower-half represents Taxonomy B, which is based on the components of a typical cloud application architecture. The taxonomy has 24 sub-categories, which are structured into 8 high-level categories: four categories each for Taxonomy, A and B.

Hybrid Deployment Requirements: The thick lines (Table I) show the space occupied by patterns used for hybrid-deployment scenarios. There are two groups of hybrid-related patterns: one related to the cloud environment and the other related to the cloud-hosted application. For example, the hybrid cloud pattern (i.e., under “hybrid clouds” sub-category of Taxonomy A) is used to integrate different clouds into a homogenous environment while the hybrid data pattern (i.e., under “hybrid cloud applications” sub-category of Taxonomy B) is used to distribute the functionality of a data handling component among different clouds.

Examples of Related Patterns: Entries in the “Related Pattern” column show examples of patterns drawn from well-known collections of cloud patterns such as [7][8][22]. The cloud patterns found in these collections may have different names but they share the same underlying implementation principle. For example, message-oriented middle-ware pattern [7] is captured in Homer et al. [8] and Wilder [22] as a Queue-centric workflow pattern and competing consumers pattern, respectively.

B. GSD Tool Selection

We carried out an empirical study to find out: (1) the type of GSD tools used in large-scale distributed enterprise software development projects; and (2) what tasks they utilize the GSD tools for.

TABLE I. TAXONOMY OF DEPLOYMENT PATTERNS FOR CLOUD-HOSTED APPLICATIONS

Deployment Components	Categories of Deployment Patterns		Related Patterns
	Main Categories	Sub-Categories	
Cloud-hosted Environment (Taxonomy A)	Application Process	Project processes	Static workload
		Implementation processes	Continuously changing workload
		Support processes	Continuously changing workload
	Core Cloud Properties	Rapid Elasticity	Elastic platform, Autoscaling [22]
		Resource Pooling	Shared component, Private cloud
		Measured Service	Elastic Platform, Throttling [8]
	Cloud Service Model	Software resources	SaaS
		Platform resources	PaaS
		Infrastructure resources	IaaS
	Cloud Deployment Model	Private clouds	Private cloud
		Community clouds	Community cloud
		Public clouds	Public cloud
	Hybrid clouds	Hybrid cloud	
Composite Cloud Application	Hybrid cloud applications	Hybrid Processing, Hybrid Data, Multisite Deployment [22]	
Cloud-hosted Application (Taxonomy B)	Cloud Management	Decomposition style	2-tier/3-tier application, Content Delivery Network [22]
		Management Processes	Update Transition Process, Scheduler Agent [8]
		Management Components	Elastic Manager, Provider Adapter, External Configuration Store [8]
	Cloud Offerings	Communication Offering	Virtual Networking, Message-Oriented Middleware
		Storage Offering	Block Storage, Database Sharding [22], Valet Key [8]
		Processing Offerings	Hypervisor, Map Reduce [22]
	Cloud Application Architecture	Cloud Environment Offerings	Elastic Infrastructure, Elastic Platform, Runtime Reconfiguration [8]
		Integration	Integration Provider, Restricted Data Access Component
		Multi-tenancy	Shared Component, Tenant-Isolated Component
		Application components	Stateless Component, User Interface Component

1) *Research Site:* The study involved 8 international companies, and interviews were conducted with 46 practitioners. The study was conducted between January, 2010 and May, 2012; and then updated between December, 2013 and April, 2014. The companies were selected from a population of large enterprises involved in both on-shore and off-shore software development projects. The companies had head offices in countries spread across three continents: Europe (UK), Asia (India), and North America (USA). Data collection involved document examination/reviews, site visits, and interviews. Further details of the data collection and data analysis procedure used in the empirical study can be seen in Bass [11].

2) *Derived Dataset of GSD Tools:* The selected set of GSD tools are: JIRA [42], VersionOne [43], Hudson [44], Subversion [45] and Bugzilla [46]. We selected these tools for two main reasons: (i) Practitioners confirmed the use of these tools in large scale geographically distributed enterprise software development projects [11]; (ii) The tools represent a mixture of open-source and commercial tools that support different software development processes; and are associated with stable developer communities (e.g., Mozilla Foundation) and publicly available records (e.g., developer's websites, whitepapers, manuals). Table II (another view of the one in [11]) shows the participating companies, projects and the GSD tools they used.

TABLE II. PARTICIPATING COMPANIES, SOFTWARE PROJECTS, SOFTWARE-SPECIFIC PROCESS AND GSD TOOLS USED

Companies	Projects	Software process	GSD tool
Company A, Bangalore	Web Mail Web Calendar	Issue tracking Code integration	JIRA Hudson
Company B, Bangalore	Web Mail Web Calendar	Issue tracking Version control	JIRA Subversion
Company H, Delhi	Customer service Airline	Agile tailoring Issue tracking	VersionOne JIRA
Company D, Bangalore (Offshore Provider to Company E)	Marketing CRM	version control Error tracking	Subversion Bugzilla
Company E, London	Banking Marketing CRM	Issue tracking Agile tailoring Code Building	JIRA VersionOne Hudson

JIRA: JIRA is a bug tracking, issue tracking and project management software tool. JIRA products (e.g., JIRA Agile, JIRA Capture) are available as a hosted solution through Atlassian OnDemand, which is a SaaS cloud offering. JIRA is built as a web application with support for plugin/API architecture that allows developers to integrate JIRA with third-party applications such as Eclipse, IntelliJ IDEA and Subversion [42].

Hudson: Hudson is a Continuous Integration (CI) tool, written in Java, for deployment in a cross-platform environment. Hudson is hosted partly as an Eclipse Foundation project and partly as a Java.NET project. It has a rich set of plugins making it easy to integrate with other software tools [47]. Organizations such as Apple and Oracle use Hudson for setting up production deployments and automating the management of cloud-based infrastructure [44].

VersionOne: VersionOne is an all-in one agile management tool built to support agile development methodologies such as Scrum, Kanban, Lean, and XP [43]. It has features that support the handling of vast amounts of reports and globally distributed teams in complex projects covering all aspects of teams, backlog and sprint planning. VersionOne can be deployed as a SaaS (on-demand) or On-Premises (local) [48].

Subversion: Subversion is a free, open source version control system used in managing files and directories, and the changes made to them over time [45]. Subversion implements a centralized repository architecture whereby a single central server hosts all project metadata. This facilitates distributed file sharing [19].

BugZilla: Bugzilla is a web-based general-purpose bug tracker and testing tool originally developed and used for the Mozilla project [46]. Several organizations use BugZilla as a bug tracking system for both open source (Apache, Linux, Open Office) and proprietary projects (NASA, IBM) [49].

C. Applying the Taxonomy

In this section, we demonstrate the practicality of the taxonomy in two ways: (1) Positioning the selected GSD tools against the taxonomy; and (2) Presenting a process for identifying applicable deployment patterns for cloud deployment of GSD tools. This framework may be used for other similar GSD tools not listed in our dataset.

1) *Positioning GSD Tools on the Taxonomy*: We demonstrate the practicality of the taxonomy by applying it to position a selected set of GSD tools. We used the collection of patterns from [7] as our reference point, and then complemented the process with patterns from [8][22].

The structure of the positioned deployment pattern, in its textual form, is specified as a string consisting of three sections-(i) Applicable deployment patterns; (ii) Technologies required to support such implementation; and (iii) Known uses of how the GSD tool (or one of its products) implements or supports the implementation of the pattern. In a more general sense, the string can be represented as: [PATTERN; TECHNOLOGY; KNOWN USE]. When more than one pattern or technology is applicable, we separate them with commas. Each sub-category of the taxonomy represents a unique class of reoccurring cloud deployment problem, while the applicable deployment pattern represents the solution.

2) *How to Identify Applicable Deployment Patterns using the Taxonomy*: Based on the experience gathered from positioning the selected GSD tools on the taxonomy, we describe CLIP (CLOUD-based Identification process for deployment Patterns), a general process for guiding software architects in selecting applicable cloud deployment patterns for GSD tools using the taxonomy. The development of CLIP (shown in Figure 2 in Business Process Model and Notation (BPMN)) was inspired by IDAPO. Stol et al. [6] used IDAPO to describe a process for identifying architectural patterns embedded in the design of open-source software tools.

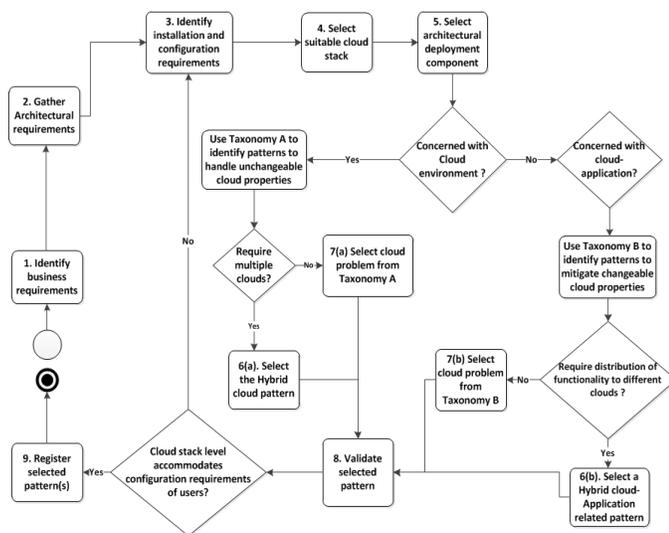


Fig. 2. CLIP Framework for Identifying Deploying Patterns

The process of selecting the cloud deployment pattern(s) is an iterative process. The first step is to (1) **find out the main business requirements of the organization**. An example of a business requirement is fast feedback time, secure access to the shared component, and even the requirement of limited resource. The next step is to (2) **gather information about the architectural structure of the GSD tool**. We recommend the use of a IDAPO, a process designed by Stol et al [6] for

identifying architectural patterns in an open-source software. At the end of that process, the architect would be able to identify among other things, the type of software and its domain, the technologies used, components and connectors, data source requirements (e.g., database type, data access method, file system etc.), and the default architectural pattern used in the design of the software.

After gathering information about the architectural structure of the GSD tool, the next step is to (3) **identify all the installation and configuration requirements of the GSD tool**. This information can be obtained directly from the documentation of the GSD tool or by creating a test application with the GSD tool. Based on the information gathered in the previous steps, the architect would be able to (4) **from the given cloud infrastructure, select a suitable level of the cloud-application stack that will accommodate all the installation and configuration requirements of the user**. If in doubt, we recommend that the architect should start with the first cloud stack level, which is the application level (i.e., GSD tool together with the software process it supports).

At this stage, the architect has to (5) **choose the architectural deployment component of interest**. In the cloud (as in other distributed environments), a cloud deployment pattern targets either the cloud environment or the cloud-application. If the architect is concerned with the cloud environment, then Taxonomy A should be used to select patterns for mapping business requirements to the unchangeable cloud properties, such as the location of the cloud infrastructure. However, if the architect is concerned with the cloud-hosted application, then Taxonomy B should be used to select deployment patterns for mitigating cloud properties, for example, performance and availability of the cloud-application.

The architect should then (6) **check for hybrid deployment requirements**. Usually, there are three main requirements that motivate the use of a hybrid-related cloud pattern. These include: (i) elasticity where there is need to increase or decrease the availability of cloud resources; (ii) accessibility; and (iii) combined assurance of privacy, security and trust [7]. For Taxonomy A, a typical requirement would be the need for integration of multiple clouds into a homogenous environment (e.g., using the hybrid cloud pattern), while that of Taxonomy B would be the need for distribution of the functionality/components of the GSD tool among different clouds (e.g., using the hybrid processing pattern). In either case, the respective hybrid related sub-category should be referenced to identify applicable patterns. otherwise the architect has to (7) **select a cloud deployment problem that corresponds to the sub-category of the chosen Taxonomy**. We have arranged the cloud deployment patterns into 8 high-level categories and 24 sub-categories that represent a recurring cloud deployment problem.

At this point, the process of selecting suitable deployment patterns involves referencing many sources of information several times. The architect can map the component/process of the GSD tool with the resources of the cloud infrastructure. We also recommend that the architect should revisit steps 1, 2, and 3. Assuming an architect wants Hudson to communicate with other external components/applications, then a better deployment pattern of choice would be Virtual Networking (via self service interface) to allow different users to be

isolated from each other, to improve security and shield users from performance influence. However, if the communication is required internally to exchange messages between application components, then a message-oriented middleware would be the obvious choice.

After selection, the (8) **patterns have to be validated** to ensure that the chosen cloud stack level can accommodate all the installation and configuration requirements of the GSD tool. This can be done by mapping the components/process of the GSD tool identified from the previous steps to the available cloud resources. Another option would be to create a test application with the GSD tool to check if deploying to the cloud is workable. If validation fails, the architect may move one level lower in the cloud stack and repeat the process from step 4. Once confirmed, the (9) **selected pattern(s) (together with the use case that gave rise to the selection) should be registered** in a repository for later use by other architects.

D. Validation of the Taxonomy

We validate the taxonomy in theory by adopting the approach used by Smite et al. [28] to validate his proposed taxonomy for terminologies in global software engineering. A taxonomy can be validated with respect to completeness by benchmarking against existing classifications and demonstrating its utility to classify existing knowledge [28].

We have benchmarked Taxonomy A to existing classifications: the ISO/IEC 12207 taxonomy of software life cycle processes and the components of a cloud model based on NIST cloud computing definition, NIST SP 800-145. Taxonomy B is benchmarked to components of a cloud application architecture such as cloud offering and cloud management, as proposed by Fehling et al. [7]. The collection of patterns in [7] captures all the major components and processes required to support a typical cloud-based application, such as cloud management and integration.

We demonstrate the utility of our taxonomy by: (i) positioning the 5 selected GSD tools within the taxonomy; and (ii) applying CLIP to guide an architect in identifying applicable deployment patterns together with the supporting technologies for deploying GSD tools to the cloud. Tables III and IV show that several deployment patterns (chosen from 4 studies) can be placed in the sub-categories of our taxonomy. In Section III C, we describe CLIP and then demonstrate its practicality with a motivating cloud deployment problem.

E. Case Study: Selecting Applicable Patterns for Deploying Components for Automated Build Verification Process

In this section, we present a simple case study of a cloud deployment problem to illustrate how to use the process described in this paper (i.e., CLIP) given our taxonomy to guide in the selection of applicable pattern.

Motivating Problem: A cloud deployment architect intends to deploy a data-handling component to the cloud so that its functionality can be integrated into a cloud-hosted Continuous Integration System (e.g., Hudson). The laws and regulations of the company make it liable to archive builds of source code once every week and keep it accessible for auditing purposes.

Access to the repository containing the archived source code shall be provided solely to certain groups of users. How can we deploy a single instance of this application to the cloud to serve multiple users, so that the performance and security of a particular user does not affect other users when there is a change in the workload?

Proposed Solution: In the following, we will go through the steps outlined in Section III C in order to select an appropriate cloud deployment pattern for handling the above cloud deployment problem.

Step 1: The key business requirements of this company are: (i) the shared repository that archives the source code cannot be shared; (ii) a single instance of this application should be deployed to the cloud to serve multiple users, and (iii) isolation among individual users should be guaranteed.

Step 2: Hudson is a web-based application and so it can easily be modified to support a 3-tier architectural pattern. An important component of this architectural pattern is the shared repository containing the archived data.

Step 3: Information obtained from Hudson documentation suggests that Hudson needs a fast and reliable communication channel to ensure that data is archived simultaneously between different environments/clouds.

Step 4: A review of the hardware and software requirements from Hudson documents suggests that having access to the application level and middle-level of the application stack will be sufficient to provide the configuration requirements for deploying and running Hudson on the given cloud infrastructure. A self-service interface can be provided as a PaaS (e.g., Amazon's Elastic Beanstalk) for configuring the hardware and software requirements of Hudson.

Step 5: The architectural deployment component of interest is the cloud-application itself, since the user has no direct access to the cloud IaaS. Therefore, the architect has to select a deployment pattern that can be implemented at the application level to handle the business requirements of the company. Based on this information, we turn to Taxonomy B, which contain cloud patterns used to mitigate cloud properties such as performance on the application level. Also, the fact that we are not attempting to integrate two cloud environments further strengthens the choice of our architectural deployment component of interest.

Step 6: After a careful review of the requirements, we conclude that a hybrid-related deployment pattern is the most suitable cloud deployment pattern for addressing the requirements of the customer. We assume that the data archived by Hudson contains the source code that drives a critical function of an application used by the company. Any unauthorized access to it can be disastrous to the company. The hybrid backup deployment pattern seems to be the most appropriate in this circumstance. This pattern can be used to extract and archive data to the cloud environment. Fehling et al. [7] discussed several types of hybrid -related patterns that can be used at the application level.

Step 7: As we have selected the hybrid backup pattern in the previous step, carrying out step 7 to select a deployment problem that corresponds to a particular sub-category

of the taxonomy is no longer relevant. However, there are other patterns that can be selected from Taxonomy B for complementary purposes. For example, in a situation where the performance of the communication channel is an issue, the message-oriented pattern can be used to assure the reliability of messages sent from several users to access the component that is shared.

Step 8: The selected deployment pattern was validated by carefully reviewing its implementation to ensure that it can accommodate the user's configuration requirements and ultimately address the cloud deployment problem. We mapped Hudson and its supporting components to the available cloud resources.

In Figure 6, we show the architecture of the hybrid backup that is proposed for solving the cloud deployment problem. The architecture consists of two environments: one is a static environment that hosts Hudson and the other is an elastic cloud environment where the cloud storage offering (e.g., Amazon's S3) resides. This static environment represents the company's Local Area Network (LAN) that runs Hudson. During Hudson's configuration on the "Post-build Action" section, the location of the files to archive should point to the storage offering that resides on the cloud environment.

The cloud storage (accessed via a REST API) has to be configured in such a way that guarantees isolation among the different users. We assume that the data handling component is initially available as a shared component for all users. To ensure that the archived data is not shared by every user, the same instance of the shared component can be instantiated and deployed exclusively for a certain number of users.

From implementation standpoint, all user-id's associated with each request to Hudson are captured and those requests with exclusive access rights are then routed to the cloud storage. We discuss an approach named "COMITRE (Cloud-based approach to Multitenancy Isolation Through request RE-routing)" in Ochei et al. [50] for deploying a single instance of Hudson to the cloud for serving multiple users (i.e., multi-tenancy) in such a way that guarantees different degrees of isolation among the users.

The different degrees of isolation between users accessing an application component that is shared is captured in three multitenancy patterns: shared component, tenant-isolated component and dedicated component [7].

Step 9: Finally, the cloud deployment scenario, the selected patterns together with the implemented architecture is documented for reference and reuse by other architects.

IV. FINDINGS

In this section, we present the findings obtained by applying the taxonomy against a selected set of GSD tools: JIRA, VersionOne, Hudson, Subversion and Bugzilla. Refer to Section III- B for details of the processes supported by these tools.

A. Comparing the two Taxonomies

The cloud deployment patterns featured in Taxonomy A (i.e., upper part of Table I) relate to the cloud environment

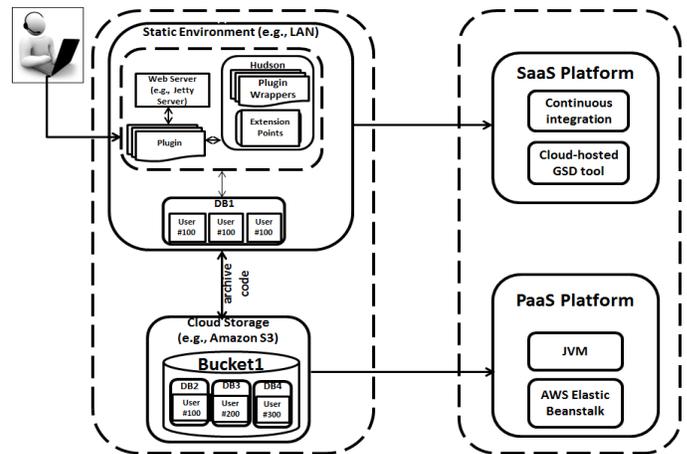


Fig. 3. Mapping Hudson to Cloud Stack based on Hybrid Backup pattern

hosting the application, while the cloud deployment patterns in Taxonomy B (i.e., lower part of Table I) relates to the cloud-hosted application itself. For example, the PaaS pattern is used to provide an execution environment to customers on the provider-supplied cloud environment. The Elastic platform pattern can be used in the form of a middleware integrated into a cloud-hosted application to provide an execution environment.

B. Hybrid-related deployment Patterns

Both taxonomies contain patterns for addressing hybrid deployment scenarios (i.e., the space demarcated with thick lines). For example, a hybrid cloud (Taxonomy A) integrates different clouds and static data centres to form a homogeneous hosting environment, while hybrid data (Taxonomy B) can be used in a scenario where data of varying sizes generated from a GSD tool resides in an elastic cloud and the remainder of the application resides in a static environment.

C. Patterns for Implementing Elasticity

We have observed that there are patterns that can be used by GSD tools to address rapid elasticity at all levels of the cloud stack. For example, an Elastic manager can be used at the application level to monitor the workload experienced by the GSD tool and its components (based on resource utilization, number of messages exchanged between the components, etc.) in order to determine how and when to provision or de-provision resources. Elastic platform and Elastic infrastructure can be used at the platform and infrastructure resources level, respectively.

D. Accessing Cloud Storage

The data handling components of most GSD tools are built on block storage architectures (e.g., relational databases such Oracle and MySQL used within Hudson and Bugzilla) for storing data, which are directly accessible by the operating system. However, a vast majority of storage offerings available on the cloud are based on object storage architecture. For example, Amazon S3, Google Cloud Storage and Windows Azure Blob provide cloud storage to cloud applications according to blob storage pattern [7]. Blob storage can be very

useful for archiving large data elements (e.g., video, installers, and ISO images arising from Hudson builds and test jobs.

E. Positioning of GSD tools on the Taxonomy

Tables II and III show the findings obtained by positioning the cloud-hosted GSD tools on each sub-category of the taxonomy. In the following, we present a shortlist of these findings to show that we can identify applicable deployment patterns to address a wide variety of deployment problems.

(i) All the GSD tools considered in this study are based on web-based architecture. For example, Bugzilla and JIRA are designed as a web-based application, which allows for separation of the user interface, and processing layers from the database that stores details of bugs/issues being tracked.

(ii) All the GSD tools support API/Plugin architecture. For example, JIRA supports several APIs that allows it to be integrated with other GSD tools. The *Bugzilla:Web services*, a standard API for external programs to interact with Bugzilla, implies support for stateless pattern. These APIs represent known uses of how these deployment patterns are implemented.

(iii) Virtualization is a key supporting technology used in combination with other patterns to achieve elasticity at all levels of the cloud stack, particularly in ensuring fast provisioning and de-provisioning of infrastructure resources.

(iv) The GSD tools use Web services (through a REST API in patterns such as integration provider [7]) to hold external state information, while messaging technology (through message queues in patterns such as Queue-centric workflow [22] and Queue-based load leveling [8]) is used to exchange information asynchronously between GSD tools/components.

(vi) Newer commercial GSD tools (JIRA and VersionOne) are directly offered as SaaS on the public cloud. On the other hand, older open-source GSD tools (Hudson, Subversion and Bugzilla) are the preferred for private cloud deployment. They are also available on the public cloud, but by third party cloud providers.

We summarize our findings as follows: Although there are a few patterns that are mutually exclusive (e.g., stateless versus stateful components, and strict versus eventual consistency [7]), most patterns still have to be combined with others (e.g. combining PaaS with Elastic platform). These deployment patterns may also use similar technologies such as REST, messaging and virtualization to facilitate their implementation.

V. DISCUSSION

The findings clearly suggest that by positioning a set of GSD tools on our proposed taxonomy, the purpose of the study has been achieved. The overarching result of the study is that most deployment patterns have to be combined with others during implementation. The findings presented here support previous research suggesting that most patterns are related and so two or more patterns can be used together [5][21].

A. Combining Related Deployment Patterns

Many deployment patterns are related and cannot be fully implemented without being combined with other ones, especially to address hybrid deployment scenarios. This scenario is very common in collaborative GSD projects, where a GSD tool either requires multiple cloud deployment environments or components, each with its own set of requirements. Our taxonomy, unlike others [22][8], clearly shows where to look for hybrid-related deployment patterns (i.e., the space demarcated by thick lines in Table I) to address this challenge. For example, when using Hudson there is usually a need to periodically extract the data it generates to store in an external storage during continuous integration of files. This implies the implementation of a hybrid data pattern. Hudson can be used in combination with other GSD tools, such as Subversion (for version control) and Bugzilla (for error tracking) within a particular software development project, each of which may also have their own deployment requirements.

B. GSD Tool Comparison

The taxonomy gives us a better understanding of various GSD tools and their cloud specific features. While other taxonomies and classifications use simple web applications [22] to exemplify their patterns, we use a mixture of commercial and open-source GSD tools. For example, commercial GSD tools (i.e., JIRA and VersionOne) are offered as a SaaS on the public cloud and also have a better chance of reflecting the essential cloud characteristic. Their development almost coincides with the emergence of cloud computing, allowing new features to be introduced into revised versions. The downside is that they offer less flexibility in terms of customization [52].

On the other hand, open-source GSD tools (i.e., Hudson, Subversion) are provided on the public cloud by third party providers and they rely on API/plugins to incorporate support for most cloud features. The downside is that many of the plugins available for integration are not maintained by the developer's community and so consumers use them at their own risk. The taxonomy also revealed that open-source GSD tools (e.g., Hudson, Subversion) are used at a later stage of a software life-cycle process in contrast to commercial tools, which are used at the early stages.

C. Support for API/Plugin Architecture

Another interesting feature of our taxonomy is that by positioning the selected GSD tools on it, it was discovered that the support for the implementation of most deployment patterns is practically achieved through API/Plugin integration [36]. This is no coincidence, as a typical cloud application is composed of various web-based related technologies such as web services, SOA and n-tier architectures. Therefore, a GSD tool with little or no support for APIs/Plugins is unlikely to attract interest from software developers. For example, JIRA's Elastic Bamboo support for Blob storage on Windows Azure is through an API [42]. JIRA has a plugin for integrating with Hudson, Subversion and Bugzilla [42] and vice versa.

D. Components and Connectors for maintaining state information and exchanging information asynchronously

Our taxonomy also highlights the technologies used to support the software processes of GSD tools, unlike others, which

TABLE III. POSITIONING GSD TOOLS ON THE PROPOSED TAXONOMY (TAXONOMY A)

Category	Sub-Category	JIRA	VersionOne	Hudson	Subversion	Bugzilla
Application Process	Project processes	Static workload, Continuously changing workload; SaaS; JIRA used by small no. of users, issues tracked reduces over time[42]	Static workload; SaaS; VersionOne is installed for a small number of users[43]	Process not supported	Process not supported	Process not supported
	Implementation processes	Process not supported	Process not supported	Continuously changing workload; PaaS; Hudson builds reduces gradually as project stabilizes[44]	Process not supported	Process not supported
	Support processes	Process not supported	Process not supported	Process not supported	Static workload, Continuously changing workload; PaaS, Hypervisor; rate of code files checked into Subversion repository is nearly constant or reduces over time[45]	Continuously changing workload; PaaS, Hypervisor; Errors tracked using Bugzilla reduces over time[46]
Core Cloud Properties	Rapid Elasticity	Stateless pattern, Elastic platform; REST API; JIRA is installed in cloud as SaaS[42]	Stateless pattern, Elastic platform; REST API; VersionOne is installed in cloud as SaaS[43]	Elastic infrastructure, shared component; hypervisor; Hudson server is supported by hypervisor in a private cloud[44]	Elastic infrastructure, tenant-isolation component; hypervisor; Subversion repository is supported by Elastic infrastructure[45]	Stateless pattern; REST API; Bugzilla is installed in cloud as SaaS in private cloud[46]
	Resource Pooling	Hypervisor, Public Cloud; ; Virtualization; JIRA deployed on the public cloud as SaaS[42]	Hypervisor, Public Cloud; Virtualization; VersionOne deployed on public cloud as SaaS[43]	Hypervisor, Tenant-isolated component; Virtualization; Hudson is deployed on a hypervisor[44]	Hypervisor, Tenant-isolated component; Virtualization; Subversion is deployed on a hypervisor[42]	Hypervisor, Public cloud; Virtualization; Bugzilla deployed on the public cloud[46]
	Measured Service	Static workload, Elastic Infrastructure, Throttling[8]; Virtualization; Small number JIRA users generates a nearly constant workload[42]	Static workload, Elastic Infrastructure, Throttling[8]; Virtualization; Small number of VersionOne users generates small workload[43]	Static workload, Elastic Infrastructure, Throttling[8]; Virtualization; Hudson can be supported on public cloud by elastic infrastructure[44]	Static workload, Elastic Infrastructure, Throttling[8]; Virtualization; Subversion can be supported on public cloud by elastic infrastructure[45]	Static workload, Elastic Infrastructure, Throttling[8]; Virtualization; Bugzilla can be supported on third party public cloud by elastic infrastructure[46]
Cloud Service Model	Software resources	SaaS; Web Services, REST; JIRA OnDemand[42]	SaaS; Web Services, REST; VersionOne OnDemand[43]	SaaS; Web Services, REST; Hudson is offered by 3 rd party cloud providers like CollabNet[51]	SaaS; Web Services, REST; Subversion is offered by 3 rd party cloud providers like CollabNet[51]	SaaS; Web Services, REST; Bugzilla is offered by 3 rd party cloud providers like CollabNet[51]
	Platform resources	PaaS; Elastic platform, Message Queuing; JIRA Elastic Bamboo[42]	PaaS; Elastic platform, Message Queuing; No known use	PaaS; Elastic platform, Message Queuing; Build Doctor and Amazon EC2 for Hudson	PaaS; Elastic platform, Message Queuing; Flow Engine powered by Jelastic for Subversion	PaaS; Elastic platform, Message Queuing; No known use
	Infrastructure resources	Not applicable	Not applicable	IaaS; Hypervisor; Hudson is a distributed execution system comprising master/slave servers[44]	IaaS; Hypervisor; Subversion can be deployed on a hypervisor	Not applicable
Cloud Deployment Model	Private usage	Private cloud; Hypervisor; JIRA can be deployed on private cloud using private cloud software like OpenStack	Private cloud; Hypervisor; VersionOne On-premises[43]	Private cloud; Hypervisor; Hudson can be deployed on private cloud using private cloud software	Private cloud; Hypervisor; Subversion can be deployed on private cloud using private cloud software	Private cloud; Hypervisor; Bugzilla can be deployed on private cloud using private cloud software
	Community usage	Community cloud; SaaS; Bugzilla can be deployed on private cloud	Community cloud; SaaS; Bugzilla can be deployed on community cloud	Community cloud; SaaS, PaaS, IaaS; Bugzilla can be deployed on community cloud	Community cloud; SaaS, IaaS; Bugzilla can be deployed on community cloud	Community cloud; SaaS, PaaS; Bugzilla can be deployed on community cloud
	Public usage	Public cloud; SaaS; JIRA OnDemand is hosted on public cloud[42]	Public cloud; SaaS; VersionOne is hosted on public cloud[43]	Public cloud; SaaS, PaaS, IaaS; Hudson is hosted on public cloud(via 3 rd party providers)[51]	Public cloud; SaaS, IaaS; Subversion is hosted on public cloud(via 3 rd party providers)[51]	Public cloud; SaaS, PaaS; Bugzilla is hosted on public cloud(via 3 rd party providers)[51]
	Hybrid usage	Hybrid cloud; SaaS; JIRA used to track issues on multiple clouds	Hybrid cloud; SaaS; Agile projects are stored in different clouds[45]	Hybrid cloud; SaaS, PaaS, IaaS; Hudson builds done in separate cloud	Hybrid cloud; SaaS, IaaS; Subversion repository resides in multiple clouds	Hybrid cloud; SaaS, PaaS; Bugzilla DB can be stored in different clouds

focus mostly on the design of cloud-native applications [7]. Web services (via REST) and messaging (via message queues) are the preferred technologies used by cloud deployment patterns (e.g., stateless pattern, message-oriented middleware) to interconnect GSD tools and other components. REST style is favoured by public cloud platforms. For example, JIRA's support for SOAP and XML-RPC is deprecated in favour of REST [42]. This trend is also reported in [22][36].

E. Accessing Data stored in Cloud Storage

Some GSD tools (e.g., Subversion) handle large amounts of data (images, music, video, documents/log files) depending on the nature of the software development project. This data can be stored on a cloud storage to take advantage of its ability to

scale almost infinitely and store large volumes of unstructured data. The downside is that the application code of the GSD tool has to be modified to enable direct HTTP-based REST API calls. Cloud storage's object architecture requires REST API to be either integrated as a plugin into the GSD tool or coded separately. Storing data on the cloud is invaluable in a case where the GSD tool runs on a static environment and the data it generates is to be archived on an elastic cloud.

F. Patterns for Cloud-application Versus Cloud-environment

Our taxonomy can be used to guide an architect in focusing on a particular architectural deployment component of interest - that is, either a cloud-hosted application or cloud-hosted environment. Other taxonomies [8][22] are concerned with

TABLE IV. POSITIONING GSD TOOLS ON THE PROPOSED TAXONOMY (TAXONOMY B).

Category	Sub-Category	JIRA	VersionOne	Hudson	Subversion	Bugzilla
Application Architecture	Application Components	User interface component, Stateless; REST API, AJAX; State information in JIRA thru REST API[42]	User-interface component, Stateless; jQuery AJAX, REST/Web Service; VersionOne REST API[43]	User-interface component, Stateless; REST API, AJAX; Hudson Dashboard pages via REST[44]	User-interface component, Stateless; REST API, AJAX; RESTful Web Services used to interact with Subversion Repositories [45]	Stateless; Bugzilla:WebService API; Bugzilla::WebService API[46]
	Multitenancy	Shared component; Elastic Platform, Hypervisor; JIRA login system[42]	Shared component; Hypervisor; VersionOne supports re-useable configuration schemes[43]	Shared component; Hypervisor; Hudson 3.2.0 supports multi-tenancy with Job Group View and Slave isolation[44]	Tenant Isolated component; Hypervisor; Global search/replace operations are shielded from corrupting subversion repository.[45]	Shared component; Hypervisor; Different users are virtually isolated within Bugzilla DB[46]
	Cloud Integration	Restricted Data Access component, Integration provider; REST API; JIRA REST API is used to integrate JIRA with other applications[42]	Integration provider; REST, Web Services; VersionOne OpenAgile Integrations platform, REST Data API for user stories[43]	Integration provider; REST, Web Services; Stapler component of Hudson's architecture uses REST[44]	Integration provider; REST, Web Services; Subversion API[45]	Integration provider; REST, Web Services; Bugzilla:WebService API[46]
Cloud Offering	Cloud environment Offering	Elastic platform; PaaS; JIRA Elastic Bamboo runs builds to create instances of remote agents in the Amazon EC2[42]	Integration provider; REST, Web Services; VersionOne's Project Management tools are used with TestComplete for automated testing environment [43]	Elastic Infrastructure/Platform, Node-based Availability; PaaS, IaaS; Hudson is a distributed build platform with "master/slave" configuration [44]	Elastic platform; PaaS; Subversion repository can be accessed by a self-service interface hosted on a shared middleware	Elastic Platform; PaaS; Bugzilla s hosted on a middleware offered by providers[46]
	Processing Offering	Hypervisor; Virtualization; JIRA is deployed on virtualized hardware	Hypervisor; Virtualization; VersionOne can be deployed on virtualized hardware	Hypervisor; Virtualization; Hudson is deployed on virtualized hardware	Hypervisor; Virtualization; Subversion is deployed on virtualized hardware	Hypervisor; Virtualization; Bugzilla is deployed on virtualized hardware
	Storage Offering	Block; Virtualization; Elastic Bamboo can access centralized block storage thru an API integrated into an operating system running on virtual server[42]	Block storage; Virtualization; VersionOne can access centralized block storage thru an API integrated into an operating system running on virtual server[43]	Block, Blob storage; Virtualization; Azure Blob service used as a repository of build artifacts created by a Hudson	Hypervisor; Virtualization; Subversion can access centralized block storage thru an API integrated into an operating system running on virtual server	Hypervisor; Virtualization; Bugzilla can access centralized block storage thru an API integrated into an operating system running on virtual server
	Communication Offering	Message-Oriented Middleware; Message Queuing; JIRA Mail Queue[42]	Message-Oriented Middleware; Message Queuing; VersionOne's Defect Work Queues[43]	Message-Oriented Middleware, Virtual networking; Message Queuing, Hypervisor; Hudson's Execution System Queuing component	Message-Oriented Middleware; Message Queuing; Subversion's Repository layer[45]	Message-Oriented Middleware; Message Queuing; Bugzilla's Mail Transfer Agent[46]
Cloud Management	Management Components	Provider Adapter, Managed Configuration, Elastic manager; RPC, API; JIRA Connect Framework[42], JIRA Advanced configuration	Managed Configuration; RPC, API; VersionOne segregation and appl. configuration	Elastic load balancer, watchdog; Elastic platform; Hudson execution system's Load Balancer component)	Managed Configuration; RPC, API; configuration file is used to configure how/when builds are done	Managed Configuration; RPC, API; Bugzilla can use configuration file for tracking and correcting errors
	Management Processes	Elastic management process; Elasticity Manager; JIRA Elastic Bamboo, and Time Tracking feature[42]	Elastic management process; Elasticity Manager; VersionOne's OnDemand security platform[43]	Update Transition process; Message Queuing; continuous integration of codes by Hudson's CI server[44]	Update Transition process; Message Queuing; continuous updates of production versions of the appl. by Subversion[45]	Resiliency management process; Elasticity platform; Bugzilla Bug monitoring/reporting feature[46]
Composite Application	Decomposition Style	3-tier; stateless, processing and data access components; JIRA is web-based application[42]	3-tier; stateless, processing and data access components; VersionOne is a web application[43]	3-tier, Content Dist. Network; user interface, processing, data access components, replica distr.; Hudson is an extensible web application, code file replicated on multiple clouds[44]	3-tier; stateless, processing and data access components; Subversion is a web-based application [45]	3-tier; stateless, processing and data access components; Bugzilla is a web application[46]
	Hybrid Cloud Application	Hybrid processing; processing component; JIRA Agile used to track daily progress work[42]	Hybrid Development Environment; processing component; VersionOne's OpenAgile Integration[43]	Hybrid Data, Hybrid Development Environment; data access component; Separate environment for code verification and testing	Hybrid Data, Hybrid Backup; data access component, stateless; Code files extracted for external storage	Hybrid Processing; processing component; DB resides in data center, processing done in elastic cloud

the design of cloud-native applications. Assuming an architect is either interested in providing the right cloud resources, or mapping the business requirement to cloud properties that cannot be changed (e.g., location and ownership of the cloud infrastructure), then Taxonomy A would be more relevant.

However, if the interest is in mitigating certain cloud properties that can be compensated at an application level (e.g., improving the availability of the cloud-hosted GSD tool), then Taxonomy B should be considered. Fehling et al. describe other cloud properties that are either unchangeable or compensatable for deploying cloud applications [7].

VI. RECOMMENDATIONS

In this section, we present a set of recommendations in the form of selection criteria in Table V to guide an architect in choosing applicable deployment patterns for deploying any GSD tool.

To further assist the architect in making a good choice, we describe CLIP (CLOUD-based Identification process for deployment Patterns), a general process for guiding architects in selecting applicable cloud deployment patterns for GSD tools using our taxonomy. The development of CLIP was

TABLE V. RECOMMENDATIONS FOR SELECTING APPLICABLE DEPLOYMENT PATTERNS FOR CLOUD DEPLOYMENT OF GSD TOOLS.

Category	Sub-Category	Selection Criteria	Applicable Patterns
Application Process	Project Processes	Elasticity of the cloud environment is not required	Static workload
	Implementation Processes	Expects continuous growth or decline in workload over time	Continuously changing workload
	Support Processes	Resources required is nearly constant;continuous decline in workload	Static workload, Continuously changing workload
Core Cloud Properties	Rapid Elasticity	Explicit requirement for adding or removing cloud resources	Elastic platform, Elastic Infrastructure
	Resource Pooling	Sharing of resources on specific cloud stack level-IaaS, PaaS, SaaS	Hypervisor, Standby Pooling Process
	Measured Service	Prevent monopolization of resources	Elastic Infrastructure, Platform, Throttling/Service Metering[8]
Cloud Service Model	Software Resources	No requirement to deploy and configure GSD tool	Software as a Service
	Platform Resources	Requirement to develop and deploy GSD tool and/or components	Platform as a Service
	Infrastructure as a Service	Requires control of infrastructure resources (e.g., storage, memory) to accommodate configuration requirements of the GSD tool	Infrastructure as a Service
Cloud Deployment Model	Private Usage	Combined assurance of privacy, security and trust	Private cloud
	Community Usage	Exclusive access by a community of trusted collaborative users	Community cloud
	Public Usage	Accessible to a large group of users/developers	Public cloud
	Hybrid Usage	Integration of different clouds and static data centres to form a homogenous deployment environment	Hybrid cloud
Application Architecture	Application Components	Maintains no internal state information	User Interface component, Stateless pattern
	Multitenancy	A single instance of an application component is used to serve multiple users, depending on the required degree of tenant isolation	Shared component, tenant-isolated component, dedicated component
	Integration	Integrate GSD tool with different components residing in multiple clouds	Integration provider, Restricted Data Access component
Cloud Offering	Cloud environment	Requires a cloud environment configured to suit PaaS or IaaS offering	Elastic platform, elastic infrastructure
	Processing Offering	Requires functionality to execute workload on the cloud	Hypervisor
	Storage Offering	Requires storage of data in cloud	Block storage, relational database
	Communication Offering	(1) Require exchange of messages internally between appl. components; (2) Require communication with external components	(1) Message-oriented middleware; (2) Virtual Networking
Cloud Management	Management Components	(1) Pattern supports Asynchronous access; (2) State information is kept externally in a central storage	(1) Provider Adapter; Elastic manager; Managed Configuration
	Management Processes	(1)Application component requires continuous update; (2) Automatic detection and correction of errors	(1) Update Transition process; (2) Resiliency management process
Composite Application	Decomposition Style	Replication or decomposition of application functionality/components	(1) 3-tier; (2) Content Distribution Network
	Hybrid Cloud Application	Require the distribution of functionality and/or components of the GSD tool among different clouds	(1) Hybrid processing; (2) Hybrid Data; (3) Hybrid Backup; (4) Hybrid Development Environment

inspired by IDAPO, a similar process proposed by Stol et al. for identifying architectural patterns in open source software [6]. The key for the successful use of CLIP is selecting a suitable level of cloud stack that will accommodate all the configuration requirements of the GSD tool to be selected. The architect has more flexibility to implement or support the implementation of a deployment pattern when there is greater “scope of control” of the cloud stack according to either the SaaS, PaaS or IaaS service delivery model [9]. For example, to implement the hybrid data pattern [7] for deploying Hudson to an elastic cloud, the architect would require control of the infrastructure level of the cloud stack to allow for provisioning and de-provisioning of resources (e.g., storage, memory, CPU).

VII. LIMITATIONS OF THE STUDY

There are multiple taxonomies developed by researchers to categorize the cloud computing space into various aspects such as, cloud resources provisioned to customers, features of cloud environment for research and scientific experimentation, cloud usage scenarios [53] and cloud architectural patterns. This study considered cloud deployment patterns that could be used to design and deploy applications to the cloud.

The findings of this study should not be generalized to:
 (i) all cloud hosted software services. We focused on cloud-hosted GSD tools (e.g., Hudson) used for large-scale distributed enterprise software development projects.
 (ii) small and medium size software development projects.

Large projects are usually executed with stable and reliable GSD tools. For small projects (with few developers and short duration), high performance and low cost may be the main consideration in tool selection.

The small number of GSD tools in the selected dataset is appropriate because we are not carrying out a feature-analysis based study of GSD tools, but only using it to apply against our proposed taxonomy. Future research can be done to re-evaluate how new GSD tools can be positioned within the taxonomy.

VIII. CONCLUSION

In this paper, we have created and used a taxonomy of deployment patterns for cloud-hosted applications to contribute to the literature on cloud deployment of Global Software Engineering tools.

Eight categories that form the taxonomy have been described: Application process, Cloud properties, Service model, Deployment model, Application architecture, Cloud offerings, Cloud management, and Composite applications. *Application process* contains patterns that handles the workload imposed on the cloud infrastructure by the ISO/IEC 12207 software processes. *Cloud properties* contains patterns for mitigating the core cloud computing properties of the tools. Patterns in *Service model* and *Deployment model* reflect the NIST cloud definition of service models and deployment models, respectively. *Application architectures* contains patterns that

support the architectural components of a cloud-application. Patterns in *Cloud offerings* reflect the main offerings that can be provided to users on the cloud infrastructure. *Cloud management* contains patterns used to manage both the components and processes of software tools. *Composite cloud* contains patterns that can be formed by combining other patterns or can be decomposed into separate components.

These categories were further partitioned into 24 sub-categories, which were mapped to the components of an (architectural) deployment structure. This mapping reveals two components classes: cloud-hosted environment and cloud-hosted application. Cloud-hosted environment and cloud-hosted application classes capture patterns that can be used to address deployment challenges at the infrastructure level and application level, respectively.

By positioning a selected set of software tools, JIRA, VersionOne, Hudson, Subversion and Bugzilla, on the taxonomy, we were able to identify applicable deployment patterns together with the supporting technologies for deploying cloud-hosted GSD tools. We observed that most deployment patterns are related and can be implemented by combining with others, for example, in hybrid deployment scenarios to integrate data residing in multiple clouds.

We have described CLIP, a novel approach for selecting applicable cloud deployment patterns, and thereafter applied it to a motivating deployment problem involving the cloud deployment of a GSD tool to serve multiple users in such a way that guarantees isolation among different users. We have also provided recommendations in tabular form, which shows the selection criteria to guide an architect in choosing applicable deployment patterns. Examples of deployment patterns derived from applying these selection criteria have been presented.

We plan to carry out several Case Studies involving the deployment of cloud-hosted GSD tools to compare how well different deployment patterns perform under different deployment conditions with respect to specific Global Software Development tools and processes (e.g., continuous integration with Hudson). Thereafter, we will carry out a cross-case analysis to synthesize the findings of the different case studies. In the future, we will develop a Decision Support Model based on the CLIP framework to help software architects in automating the process of selecting applicable cloud deployment patterns for GSD tools. This will speed up the cloud deployment process by providing proven patterns with the supporting technologies.

ACKNOWLEDGMENT

This research was supported by the Tertiary Education Trust Fund (TETFUND), Nigeria, and Robert Gordon University, UK.

REFERENCES

[1] L. C. Ochei, J. M. Bass, and A. Petrovski, "Taxonomy of deployment patterns for cloud-hosted applications: A case study of global software development (gsd) tools," in *The Sixth International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2015)*. IARIA, 2015, pp. 86–93.

[2] R. Buyya, J. Broberg, and A. Goscinski, *Cloud Computing: Principles and Paradigms*. John Wiley & Sons, Inc., 2011.

[3] M. A. Chauhan and M. A. Babar, "Cloud infrastructure for providing tools as a service: quality attributes and potential solutions," in *Proceedings of the WICSA/ECSA 2012 Companion Volume*. ACM, 2012, pp. 5–13.

[4] S. Junuzovic and P. Dewan, "Response times in n-user replicated, centralized, and proximity-based hybrid collaboration architectures," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. ACM, 2006, pp. 129–138.

[5] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice, 3/E*. Pearson Education India, 2013.

[6] K.-J. Stol, P. Avgeriou, and M. A. Babar, "Design and evaluation of a process for identifying architecture patterns in open source software," in *Software Architecture*. Springer, 2011, pp. 147–163.

[7] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Armitter, *Cloud Computing Patterns*. Springer, 2014.

[8] A. Homer, J. Sharp, L. Brader, M. Narumoto, and T. Swanson, *Cloud Design Patterns*, R. Corbisier, Ed. Microsoft, 2014.

[9] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, "Cloud computing synopsis and recommendations," *NIST special publication*, vol. 800, p. 146, 2012.

[10] S. Hansma, "Go fast and don't break things: Ensuring quality in the cloud," in *Workshop on High Performance Transaction Systems (HPTS 2011), Asilomar, CA, October 2011. Summarized in Conference Reports column of USENIX; login 37(1)*, February 2012.

[11] J. Bass, "How product owner teams scale agile methods to large distributed enterprises," *Empirical Software Engineering*, pp. 1–33, 2014.

[12] W. Aspray, F. Mayadas, M. Y. Vardi *et al.*, "Globalization and offshoring of software," *Report of the ACM Job Migration Task Force, Association for Computing Machinery*, 2006.

[13] J. D. Herbsleb, "Global software engineering: The future of socio-technical coordination," in *2007 Future of Software Engineering*. IEEE Computer Society, 2007, pp. 188–198.

[14] C. Larman and B. Vodde, *Practices for scaling lean and agile development: large, multisite, and offshore product development with large-scale Scrum*. Pearson Education, 2010.

[15] F. Lanubile, "Collaboration in distributed software development," in *Software Engineering*. Springer, 2009, pp. 174–193.

[16] J.-P. Pesola, H. Tanner, J. Eskeli, P. Parviainen, and D. Bendas, "Integrating early v and v support to a gse tool integration platform," in *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on*. IEEE, 2011, pp. 95–101.

[17] M. A. Babar and M. Zahedi, "Global software development: A review of the state-of-the-art (2007–2011)," IT University of Copenhagen, Tech. Rep., 2012.

[18] J. D. Herbsleb and A. Mockus, "An empirical study of speed and communication in globally distributed software development," *Software Engineering, IEEE Transactions on*, vol. 29, no. 6, pp. 481–494, 2003.

[19] F. Lanubile, C. Ebert, R. Prikladnicki, and A. Vizcaíno,

- “Collaboration tools for global software engineering,” *Software, IEEE*, vol. 27, no. 2, pp. 52–55, 2010.
- [20] J. Portillo-Rodriguez, A. Vizcaino, C. Ebert, and M. Piattini, “Tools to support global software development processes: a survey,” in *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*. IEEE, 2010, pp. 13–22.
- [21] J. Vlissides, R. Helm, R. Johnson, and E. Gamma, “Design patterns: Elements of reusable object-oriented software,” *Addison-Wesley*, vol. 49, p. 120, 1995.
- [22] B. Wilder, *Cloud Architecture Patterns*, 1st ed., R. Roumeliotis, Ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472.: O’Reilly Media, Inc., 2012.
- [23] IEEE, “Ieee standard glossary of software engineering terminology,” *Office*, vol. 121990, no. 1, p. 1, 1990.
- [24] M. Unterkalmsteiner, R. Feldt, and T. Gorschek, “A taxonomy for requirements engineering and software test alignment,” *ACM Transactions on Software Engineering and Methodology*, vol. Vol. V, No. N, Article A, 2013. [Online]. Available: <http://doi.acm.org/10.1145/0000000.0000000>
- [25] R. L. Glass and I. Vessey, “Contemporary application-domain taxonomies,” *Software, IEEE*, vol. 12, no. 4, pp. 63–76, 1995.
- [26] D. I. Sjoberg, T. Dyba, and M. Jorgensen, “The future of empirical methods in software engineering research,” in *Future of Software Engineering, 2007. FOSE’07*. IEEE, 2007, pp. 358–378.
- [27] J. Buckley, T. Mens, M. Zenger, A. Rashid, and G. Kniesel, “Towards a taxonomy of software change,” *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 17, no. 5, pp. 309–332, 2005.
- [28] D. Smite, C. Wohlin, Z. Galvina, and R. Prikladnicki, “An empirically based terminology and taxonomy for global software engineering,” *Empirical Software Engineering*, pp. 1–49, 2012.
- [29] R. Dupuis, “Software engineering body of knowledge,” 2004.
- [30] C. Moyer, *Building Applications for the Cloud: Concepts, Patterns and Projects*. Pearson Education, Inc, Rights and Contracts Department, 501 Boylston Street, Suite 900, Boston, MA 02116, USA: Addison-Wesley Publishing Company, 2012.
- [31] N. Sawant and H. Shah, *Big Data Application Architecture - A problem Solution Approach*. Apress, 2013.
- [32] Z. Mahmood, *Cloud Computing: Methods and Practical Approaches*. Springer-Verlag London, 2013.
- [33] T. Erl and A. Naserpour, *Cloud Computing Design Patterns*. Prentice Hall, 2014.
- [34] S. Strauch, U. Breitenbuecher, O. Kopp, F. Leymann, and T. Unger, “Cloud data patterns for confidentiality,” in *Proceedings of the 2nd International Conference on Cloud Computing and Service Science, CLOSER 2012, 18-21 April 2012, Porto, Portugal*. SciTePress, 2012, pp. 387–394.
- [35] J. Varia, “Migrating your existing applications to the cloud: a phase-driven approach to cloud migration.” Amazon Web Services (AWS), [Online: accessed in November, 2014 from <http://aws.amazon.com/whitepapers/>].
- [36] J. Musser. (2012) Enterprise-class api patterns for cloud and mobile. CITO Research.
- [37] Arista.com. Cloud networking: Design patterns for cloud-centric application environments. ARISTA Networks, Inc. Online: accessed in November, 2015 from <http://www.arista.com/assets/data/pdf/CloudCentric...pdf>.
- [38] C. Brandle, V. Grose, M. Young Hong, J. Imholz, P. Kaggali, and M. Mantegazza. (2014, June) Cloud computing patterns of expertise. IBM. International Technical Support Organization. [Online]. Available: ibm.com/redbooks
- [39] J. Varia. (2014) Architecting for the cloud: best practices. Amazon Web Services (AWS). Online: accessed in November, 2015 from <http://aws.amazon.com/whitepapers/>.
- [40] L. Lilien, “A taxonomy of specialized ad hoc networks and systems for emergency applications,” in *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*. IEEE, 2007, pp. 1–8.
- [41] P. Mell and T. Grance, “The nist definition of cloud computing,” *NIST special publication*, vol. 800, no. 145, p. 7, 2011.
- [42] Atlassian.com. Atlassian documentation for jira 6.1. Atlassian, Inc. Online: accessed in November, 2015 from <https://www.atlassian.com/software/jira/>.
- [43] VersionOne, “Versionone-agile project management and scrum,” VersionOne Inc., [Online: accessed in November, 2015 from <http://www.versionone.com/>].
- [44] M. Moser and T. O’Brien. The hudson book. Oracle, Inc., USA. Online: accessed in November, 2015 from <http://www.eclipse.org/hudson/the-hudson-book/book-hudson.pdf>.
- [45] B. Collins-Sussman, B. Fitzpatrick, and M. Pilato, *Version control with subversion*. O’Reilly, 2004.
- [46] Bugzilla.org. The bugzilla guide. The Mozilla Foundation. [Online: accessed in November, 2015 from <http://www.bugzilla.org/docs/>].
- [47] Hudson. Hudson extensible continuous integration server. Oracle Inc., USA. [Online: accessed in November, 2015 from <http://www.hudson-ci.org/>].
- [48] Versionone.com. Top 10 reasons to choose versionone. VersionOne Inc. [Online: accessed in November, 2015 from <http://www.versionone.com/whyversionone.pdf>].
- [49] N. Serrano and I. Ciordia, “Bugzilla, itracker, and other bug trackers,” *Software, IEEE*, vol. 22, no. 2, pp. 11–13, 2005.
- [50] L. C. Ochei, J. M. Bass, and A. Petrovski, “Evaluating degrees of multitenancy isolation: A case study of cloud-hosted gsd tools,” in *2015 International Conference on Cloud and Autonomic Computing (ICAC)*. IEEE, 2015, pp. 101–112.
- [51] CollabNet. Subversionedge for the enterprise. CollabNet, Inc. [Online: accessed in November, 2015 from <http://www.collab.net/products/subversion/>].
- [52] I. Sommerville, *Software Engineering*. Pearson Education, Inc. and Addison-Wesley, 2011.
- [53] A. Milenkoski, A. Iosup, S. Kounev, K. Sachs, P. Rygielski, J. Ding, W. Cirne, and F. Rosenberg, “Cloud usage patterns: A formalism for description of cloud usage scenarios,” Standard Performance Evaluation Corporation(SPEC) Research Cloud Working Group, Tech. Rep., 2013.

Query Interpretation – an Application of Semiotics in Image Retrieval

Maaïke H.T. de Boer^{1,2}, Paul Brandt^{1,3}, Maya Sappelli^{1,2}, Laura M. Daniele¹, Klamer Schutte¹, Wessel Kraaij^{1,2}

1: TNO
Anna van Buerenplein 1,
2595 DA The Hague,
The Netherlands
{maaïke.deboer, paul.brandt,
maya.sappelli, laura.daniele,
klamer.schutte,wessel.kraaij}@tno.nl

2: Radboud University
Toernooiveld 200
6525 EC Nijmegen
The Netherlands
{m.deboer, w.kraaij,
m.sappelli}@cs.ru.nl

3: Eindhoven University of Technology
De Groene Loper 19
Eindhoven
The Netherlands
p.brandt@tue.nl

Abstract— One of the challenges in the field of content-based image retrieval is to bridge the semantic gap that exists between the information extracted from visual data using classifiers, and the interpretation of this data made by the end users. The semantic gap is a cascade of 1) the transformation of image pixels into labelled objects and 2) the semantic distance between the label used to name the classifier and that what it refers to for the end-user. In this paper, we focus on the second part and specifically on (semantically) scalable solutions that are independent from domain-specific vocabularies. To this end, we propose a generic semantic reasoning approach that applies semiotics in its query interpretation. Semiotics is about how humans interpret signs, and we use its text analysis structures to guide the query expansion that we apply. We evaluated our approach using a general-purpose image search engine. In our experiments, we compared several semiotic structures to determine to what extent semiotic structures contribute to the semantic interpretation of user queries. From the results of the experiments we conclude that semiotic structures can contribute to a significantly higher semantic interpretation of user queries and significantly higher image retrieval performance, measured in quality and effectiveness and compared to a baseline with only synonym expansions.

Keywords— query expansion; natural language queries; image retrieval; semantic reasoning; computational semiotics.

I. INTRODUCTION

More and more sensors connected through the Internet are becoming essential to give us support in our daily life. In such a global sensor environment, it is important to provide smart access to sensor data, enabling users to search semantically in this data in a meaningful and, at the same time, easy and intuitive manner.

Towards this aim, we developed a search engine that combines content based image retrieval (CBIR), Human Media Interaction and Semantic Modelling techniques in one single application: “Google[®] for sensors” or “GOOSE” for short. This paper builds on our earlier work on applying semantic reasoning in image retrieval [1] in the GOOSE search engine, an overview paper of which is given in [2][3].

A major issue to text searches in visual data is “the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation”, coined by [4] as the

semantic gap in CBIR. Our application is able to retrieve visual data from multiple and heterogeneous sources and sensors, and responds to the fact that the semantic gap consists of two parts [5]: the first part addressing the realm where raw image pixels are transformed into generic objects to which labels are applied to represent their content; the second part addressing the realm of semantic heterogeneity, representing the semantic distance between the object labelling and the formulation by the end-user of a query that is meant to carve out a part in reality that situates that object. The GOOSE approach to closing the first section addresses image classification and quick image concept learning, presented in [6], and fast re-ranking of visual search results, presented in [7]. This paper addresses the second section of the semantic gap by applying, in this order, query parsing, concept expansion, and concept mapping to labels associated with certain classifiers. Query concepts that do not match any classifier’s label are expanded using an external knowledge base, in this case ConceptNet [8], to find alternative concepts that are semantically similar to the original query concepts but do match with a classifier label.

Whereas in [1] we only used ‘IsA’ and ‘Causes’ relations in the query expansion, in this work we address additional types of relations in order to improve the matching rate between query concepts and classifier labels. However, a drawback of considering additional relations is that the algorithms for their semantic interpretation become tightly coupled to the particular external knowledge base of choice, rendering them less applicable for other knowledge bases. To overcome this limitation and keep our semantic interpretation generically applicable to other knowledge bases, we introduce the use of semiotics that provides guidance to how humans interpret signs and how the abstract relationships between them apply. Due to its universal application, a semiotic approach not only provides us with the flexibility to use different knowledge bases than ConceptNet, but it is also independent from domain-specific terminologies, vocabularies and reasoning. By defining a simple mapping from the specific relationships of the knowledge base of choice, e.g., ConceptNet, onto semiotic structures, the semantic interpretation algorithms can latch onto the semiotic structures only. The resulting transparency between, at the one hand, the semantic interpretation algorithms and, at the other hand, abstracting from the

specifics of (i) the relationships that are available in the external knowledge base, and (ii) the domain-specific vocabularies, brings about the required general applicability of our solution.

Summarizing, we seek to improve the matching rate between query concepts and classifier labels, by 1) considering more, if not all, relations that are available in a knowledge base; while remaining 2) as independent to the external knowledgebase as possible; and 3) as computationally lean as possible.

We formulate our research question as

To what extent can semiotic structures contribute to the semantic interpretation of user queries?

In order to answer our research question, we conducted an experiment on our TOSO dataset [9], which contains 145 test images and 51 trained classifiers. Furthermore, for evaluation purposes, we defined 100 user queries [10]. We annotated these user queries with their ground truth for both parts of the semantic gap: (i) the ground truth for semantic matching, identifying the classifier labels that are meant to be found for each user query, and (ii) the ground truth for the image retrieval, identifying the images that are meant to be found. For the different types of semiotic structures we calculated the effectiveness and quality in terms of different types of F-measure for both semantic matching and image retrieval.

From the results of these experiments, we can conclude that applying semiotic relations in query expansion over an external, generic knowledge base contributes to a high quality match between query concepts and classifier labels. It also significantly improves image retrieval performance compared to a baseline with only synonym expansions. Some relations that are present in ConceptNet could not be assigned to the applied semiotic structures; inclusion of these relations in the semantic analysis provided for higher effectiveness at the cost of losing loose coupling between these relations and the algorithms that implement the semantic analysis. However, we did not investigate other potential semiotic structures to this effect.

The main contribution of this paper is a generic approach to the expansion of user queries using general-purpose knowledge bases, and how semiotics can guide this expansion independently from the specific knowledge base being used.

This paper is structured as follows: Section II describes related work on query expansion and semiotics; Section III presents a short tutorial on semiotics; Section IV provides an overview of the generic semantic interpretation system; Section V explains the semantic analysis and how we have positioned semiotic structures for its support; Section VI describes the experiment that has been performed with the application, followed by a presentation and discussion of their results in Sections VII and VIII, respectively. We conclude our work, including indications for future work, in Section IX.

II. RELATED WORK

In this section, we discuss related work in CBIR about the first part of the semantic gap, i.e., automatic classifier annotation, as well as the second part of the semantic gap, i.e., some efforts related to query expansion using semantic relations. Finally, we discuss related work on computational semiotics.

A. Automatic image annotation

Most of the effort in applying semantics in CBIR is aimed at training classifiers using large sources of visual knowledge, such as ImageNet [11] and Visipedia [12]. The trained classifiers are subsequently annotated with one or more labels that should describe their meaning. However, these annotations are often subjective, e.g., influenced by the domain of application and not accurate from a semantic point of view. Consequently, users that apply these classifiers need to have prior knowledge about the context of use of the annotations. In order to overcome this issue and facilitate the use of classifiers without the need of training, various efforts in the literature focus on improving the annotations. These efforts mainly apply domain-specific ontologies as basis for annotation, such as the ontologies in [12][13] that are used to annotate soccer games, or for the purpose of action recognition in a video surveillance scenario [15]. Although these approaches provide for more intuitive semantics that require less prior knowledge from the user, they are tailored to specific domains and cannot be reused for general-purpose applications.

B. Relation-based query expansion

Several systems proposed in the literature address query expansion exploiting relations with terms that are semantically similar to the concepts in the query [16][17][18]. The system in [16] facilitates natural language querying of video archive databases. The query processing is realized using a link parser [19] based on a light-parsing algorithm that builds relations between pairs of concepts, rather than constructing constituents in a tree-like hierarchy. This is sufficient for the specific kind of concept groups considered in the system [16], but is limitative for more complex queries.

The Never Ending Image Learner (NEIL) proposed in [17] is a massive visual knowledge base fed by a crawler that runs 24 hour a day to extract semantic content from images on the Web in terms of *objects*, *scenes*, *attributes* and their *relations*. The longer NEIL runs, the more relations between concepts detected in the images it learns. Analogously to our approach, NEIL is a general-purpose system and is based on learning new concepts and relations that are then used to augment the knowledge of the system. Although NEIL considers an interesting set of semantic relations, such as taxonomy (*IsA*), partonomy (*Wheel is part of Car*), attribute associations (*Round_shape is attribute of Apple* and *Sheep is White*), and location relations (*Bus is found in Bus depot*), most of the relations learned so far are of the basic type 'IsA' or 'LooksSimilarTo'.

Furthermore, in [18] knowledge bases ConceptNet and Wikipedia, and an expert knowledge base are compared for

semantic matching in the context of multimedia event detection. Results show that query expansion can improve performance in multimedia event detection, and that the expert knowledge base is the most suitable for this purpose. When comparing Wikipedia and ConceptNet, ConceptNet performs slightly better than Wikipedia in this field. In their comparison, the authors only considered query expansion using the ConceptNet 'IsA' relation.

C. Semiotics in CBIR

Although text analysis is its primary field of application, recently semiotics gained the interest in the field of ICT. The application of semiotics in computer science is best illustrated with the emergence of computational semiotics, where a clear starting point for its definition is the fact that signs and sign systems are central to computing: manipulation of symbols applies to everything that happens in computer science, from user interfaces [20][21][22] to software engineering [23][24][25][26], from model-driven engineering [27] to conceptual and knowledge modelling [28][29][30], and interoperability [31] alike. In relation to CBIR, many studies, summarized by [5], accept the existence of 'semantic layers' in images. Every layer provides for another abstraction and aggregation of the things that are being denoted. The studies referenced in [5] address these layers as distinct realms, and act accordingly by constraining themselves to one layer. However, semioticians address these layers as a whole, and study it as a process to which they refer as *unlimited semiosis* (see next section). We are inspired by that approach and therefore part of our work considers unlimited semiosis as algorithmic foundation when addressing these layers. Application of semiotics in CBIR and especially about user query interpretation is very limited, and the following two studies represent, to the best of our knowledge, good examples of its main focus.

Yoon [32] has investigated the association between denotative (literal, definitional) and connotative (societal, cultural) sense-making of image meta-data in support of image retrieval. This approach is similar to ours in that it is based on semiotics structures to bridge the semantic gap. Although the results are promising, it cannot be applied in our generic context due to the domain-specific foundations that are implicit to connotations.

Closely related to it, [33] studies how semiotics can account for image features that characterize an audio, visual or audio-visual object, in order to facilitate visual content description or annotation. Their model integrates low-level image features such as color and texture together with high-level denotative and connotative descriptions. This approach differs with ours in that they do not make a distinction between the two cascading parts of the semantic gap, but instead take an integrated approach.

III. SEMIOTICS

We include a brief tutorial on semiotics here since we believe this discipline is not very well known to our readers. Specifically, we address the semiotic structures that we apply. According to semiotics, humans make meanings

through our creation and interpretation of signs [34]. A sign can be anything, varying from a character to a sculpture, as long someone interprets it, i.e., it goes beyond the sign itself. A *semiotic sign*, or sign for short, represents a structure. In Peirce's semiotic triangle [35] it consists of three closely related aspects, as depicted in Figure 1(a).

The *Representamen* (sometimes denoted as the *sign vehicle*) represents the form that the sign takes, e.g., this paper or a Chinese character. This form can be written, spoken or displayed, such as a picture or movie scene. The *Interpretant* in Figure 1(a) does not refer to an interpreter but rather to the sense given to the sign, i.e., our mental representation of reality, such as the mental "picture" of a 'red apple' that one has in mind. The *Object* in Figure 1(a), is the concrete thing in reality to which the sign refers, where this reality may also be an hypothetical reality, e.g., a unicorn. As opposed to the direct relationships between the interpretant and the object, and the interpretant and the representamen, which are drawn as solid lines, the relationship between the object and representamen is not direct, and hence depicted with dots. A semiotic sign only qualifies as such when it unifies all three aspects into a meaningful ensemble: the object is perceived by our senses and abstracted into the interpretant, which subsequently is represented by the representamen. A Peircean sign concurrently indicates what is being represented, how it is being represented and how it is being interpreted. The sense making is subjective by nature, hence every actor makes use of their own signs, although the sign's representamen can be shared.

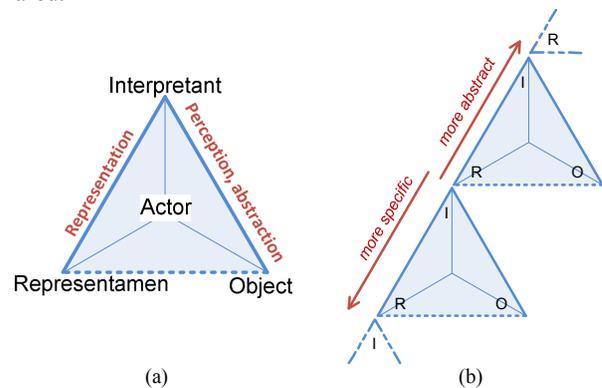


Figure 1. Peirce's semiotic triangle (a), unlimited semiosis (b).

Although semiotic techniques are used mainly to analyze texts, in this paper we extend their usage to the semantic analysis of user queries. Specifically, we consider the semiotic structures that help us to better distinguish between the relations defined in external knowledge bases, e.g., to which extent is an 'IsA' relation semiotically different from a 'Causes' relation, and how can we benefit from this difference. We selected the semiotic structures *unlimited semiosis*, *paradigms* and *syntagms* from [34] as vehicles to a universal approach towards reasoning over different semantic relationships defined in various knowledge bases.

A. Unlimited semiosis

Sense making is above all a process, and Peirce refers to the interaction between the three elements of the semiotic triangle as ‘semeiosis’ [21][22]. He also observes that semiotic signs are coupled: “a sign (...) creates in the mind of that person an equivalent sign, or perhaps a more developed sign.” (ibid.). Consequently, the interpretant at level N is yet another representamen but at a ‘more developed’ level N+1. Eco [36] uses the term ‘unlimited semiosis’ to refer to the succession of cascading signs that emerge from that, ad infinitum (Figure 1(b)). The application of unlimited semiosis gives us the capability to address semantic issues that relate to the different levels of *granularity* between the query of concept and a classifier label, e.g., ‘vehicle’ (higher level of detail) and ‘car’ (lower level of detail).

B. Paradigms

From a semiotic perspective, semantics arise from the differences between signs. In other words, without their ability to signify differences, signs could not carry meaning at all. Differences of signs concern two distinctions, as depicted in Figure 2. The first distinction, called *paradigms*, concern substitution and signify functional contrasts, e.g., how to differentiate the sentences ‘the man cried’ from ‘the woman cried’. Signs are in paradigmatic relation when the choice of one (‘man’) excludes the choice of another (‘woman’) [37], i.e., disjunction. The selection of a particular sign from a paradigmatic set, e.g., selecting *man* from {*man*, *woman*, *child*}, implies an intentional exclusion of the interpretations that originate from the use of the other signs from the paradigmatic set, e.g. *woman* and *child*. Other paradigms in Figure 2 are {*cry*, *sing*, *mutter*}, and one about cardinality. Due to their nature, paradigms provide us with the ability to reject alternative concepts.



Figure 2. The semantics of a sign is determined by both its paradigmatic and syntagmatic relations.

C. Syntagms

Where paradigms reflect differences concerning substitution, the second axis in Figure 2, *syntagms*, reflect differences concerning position, e.g., the position of words in a sentence, or the position of paragraphs in a section. This reflects how the juxtaposition of the distinct parts (the signs) complete into a whole, e.g., how words take their place in a grammatically correct sentence, or how chapters are used to form a book. Syntagmatic relations reflect the admissible combinations of paradigmatic sets into well-formed structures, e.g., conjunction. The use of one syntagmatic

structure over another influences semantics, e.g., ‘the ship that banked’ versus ‘the bank that shipped’ use identical signs, whilst their positions in the sentence turn them from a verb to an object and vice versa, with completely different semantics as result.

IV. GENERIC SEMANTIC REASONING SYSTEM

Figure 3 shows an overview of the semantic reasoning parts of the GOOSE system in which green and blue parts represent the components that realize the semantic reasoning, yellow parts represent the components dedicated to the image classification task and the white parts represent external components. The image classification task, which is elaborated in [6], captures the semantics of visual data by translating the pixels from an image into a content description (which could be a single term), coined as *annotated images*.

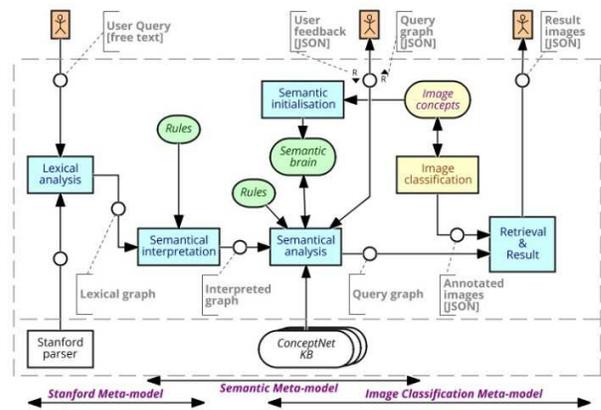


Figure 3. System overview.

The semantic reasoning starts with a user query in natural language. The query is processed by four modules, while a fifth module takes care of initializing the system and learning new concepts. In the first stage, the query is sent to the *Lexical Analysis* module that parses it using the Stanford Parser [38]. The Stanford Parser returns a lexical graph, which is used as input to the *Semantic Interpretation* module. In this module, a set of rules is used to transform the lexical elements of the Stanford meta-model into semantic elements of an intermediary ontology (our meta-model, discussed in section IV.C below). The interpreted graph is sent to the *Semantic Analysis* module that matches the graph nodes against the available image concepts. If there is no exact match, the query is expanded using an external knowledge base, i.e., ConceptNet, to find a close match. The interpretation resulting from the Semantic Analysis is presented as a query graph to the user. The query graph is also used as input for the *Retrieval and Result* module, which provides the final result to the user. In the following subsections the complete process is described in detail using the sample query *find a red bus below a brown animal*. In this particular query, its positional part, e.g., *below*, should be understood from the viewpoint of the user posing the

query, i.e., the relative positions of the ‘red bus’ and the ‘brown animal’ as shown at the user’s screen.

A. Semantic Initialization

This module provides an initial semantic capability by populating the Semantic Brain, which holds all *image concepts* that are known to the system. Image concepts are represented as instances of the meta-model (discussed in Section IV.C), and refer to those things that the image classification task is capable of detecting. This component also handles updates to the Semantic Brain following from new or modified image classification capabilities and semantic concepts.

B. Lexical Analysis

In the Lexical Analysis module, the user query is lexically analyzed using the Typed Dependency parser (englishPCFG) of Stanford University [38]. Before parsing the query, all tokens in the query are converted to lower case. In the example of *find a red bus below a brown animal*, the resulting directed graph from the Lexical Analysis is shown in Figure 4.

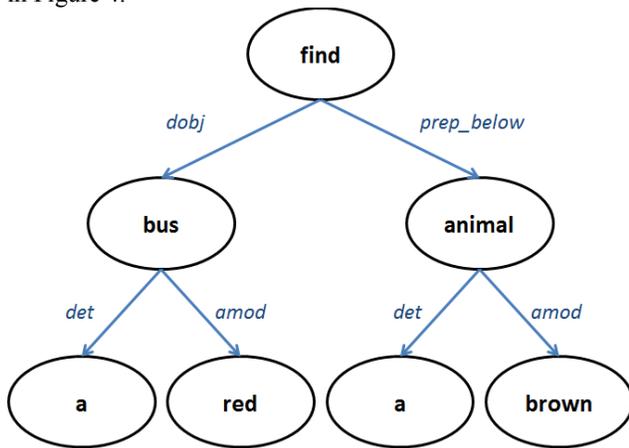


Figure 4. Lexical Graph.

C. Semantic Interpretation

Since GOOSE is positioned as a generic platform, its semantics should not depend on, or be optimized for, the specifics of one single domain of application. Instead, we apply a generic ontological commitment by defining a semantic meta-model, shown in Figure 5, which distinguishes objects that might (i) bear attributes (*a yellow car*), (ii) take part in actions (*a moving car*), (iii) occur in a scene (*outside*), and (iv) have relations with other objects, in particular ontological relations (*a vehicle subsumes a car*), spatial relations (*an animal in front of a bus*), and temporal relations (*a bus halts after driving*).

In the Semantic Interpretation module, a set of rules is used to transform the elements from the lexical graph into *objects*, *attributes*, *actions*, *scenes* and *relations*, according to the semantic meta-model in Figure 5.

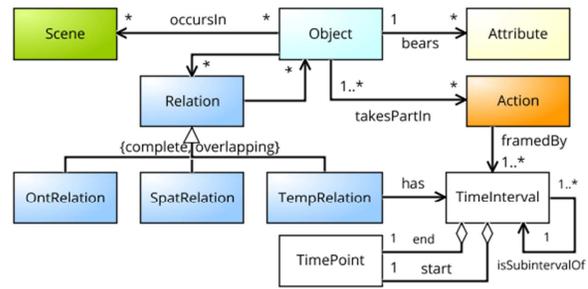


Figure 5. Semantic meta-model.

These rules include the following examples:

- Derive *cardinality* from a *determiner* (*det* in Figure 4), e.g., *the* in a noun in the singular form indicates a cardinality of 1, while *a/an* indicates at least 1;
- Derive *attributes* from *adjectival modifiers* (*amod* in Figure 4), i.e., adjectival phrases that modify the meaning of a noun;
- Derive *actions* from *nominal subjects* and *direct objects* (*nsubj* and *dobj*, absent in Figure 4), i.e., the subject and object of a verb, respectively;
- Actions that represent the query command, such as *find*, *is*, *show* and *have*, are replaced on top of the tree by the subject of the sentence.

The output of the Semantic Interpretation for the sample query *find a red bus below a brown animal* is shown in Figure 6.

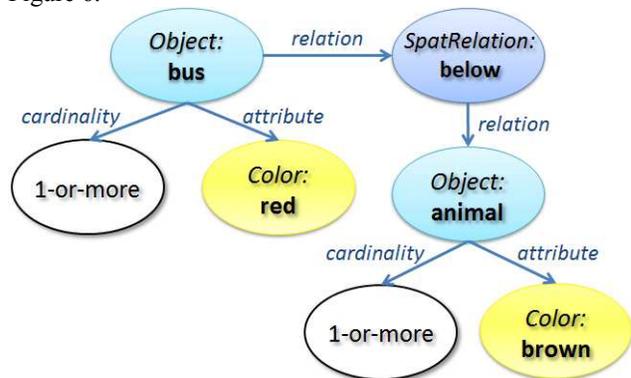


Figure 6. Interpreted Graph.

D. Semantic Analysis

The purpose of the Semantic Analysis is to align the elements from the interpreted graph, which are the query concepts, with the image concepts that are available in the Semantic Brain. For those objects, actions, scenes or attributes from the graph that do not have a syntactical identical counterpart (‘exact match’) in the Semantic Brain, and hence cannot be recognized by the image classification component, the query concepts are expanded into *alternative concepts* using an external general-purpose knowledge base. We use the external knowledge base ConceptNet to find these alternative concepts. Our principle of genericity and

loose coupling, however, facilitates the use of other or even more knowledge bases without the need to adapt the semantic analysis algorithms.

E. Retrieval and Result

This module retrieves the images that, according to the classifiers, contain concepts that carry an identical label as the query concepts (or alternative concepts). Furthermore, the cardinality, attribute and spatial relations should match with the query. If the image contains too many instances, the image is still included, however, with lower ranking. The spatial relations are determined by the edges of the bounding box. Because our bounding boxes are not accurate, we use a relaxed version of the prepositions. The upper left edge of the bounding box has the values [0,0]. In the preposition *left of*, the left edge of the bounding box of the right object should be right of the left edge of the bounding box of the left object, denoted as:

<i>Left of</i>	$a.min.x < b.min.x$
<i>Right of</i>	$a.max.x > b.max.x$
<i>On top of</i>	$a.min.y < b.min.y$
<i>Below</i>	$a.max.y > b.max.y$
<i>And</i>	$a \text{ and } b$

V. SEMIOTIC STRUCTURES IN SEMANTIC MATCHING

In this section, we explain how the semiotic structures introduced in Section II are used to implement the Semantic Analysis module of the GOOSE system.

We consider the external knowledge base, e.g., ConceptNet, to represent a graph. Query expansion is then similar to a graph traversal, where nodes represent alternative concepts, and edges represent relations between concepts. Each edge is of a specific type, e.g., *IsA*, *HasA*, *PartOf*, and more. Considering edges of particular types only, results in considering a subgraph. We apply a semiotic structure by only considering edges which type can be considered to represent that semiotic structure. This results in a subgraph, the characteristics of which corresponds to a large extend to the semiotic structure of choice. For example, by considering paradigmatic relations only, a paradigmatic subgraph emerges. Query expansion now becomes *semiotic subgraph* traversal, and will apply distinct traversal strategies that fit best the particular semiotic subgraphs. Additionally, we can also apply results from one subgraph traversal for traversals of other subgraphs, since their semiotic structures can be put into a specific relation to each other.

Whether an external knowledge base indeed will show these emerging semiotic subgraphs cannot be enforced, hence deviations should be anticipated. For instance, our algorithms will need to take into account the presence of loops in the subgraph, despite the fact that a semiotic theory might predict the emergence of a non-cyclic graph. The following sections will elaborate on the specifics for each semiotic structure.

A. Unlimited semiosis

This structure discerns relations that bear a direction-oriented application during query concept expansion, i.e., a relation expresses direction towards more *abstract* or more

specific concepts. Using unlimited semiosis, we are able to select a certain expanded concept that is either more abstract, or more specific than the original query concept. This has consequences for the corresponding classifier. The selection of a more abstract alternative concept implies that the corresponding classifier will have less granularity and is therefore more general but less accurate from a semantical point of view. In other words, the *unlimited semiosis subgraph* that emerges, represents a directed, non-cyclic multigraph. For example, requesting for a 'flower' returns the 'plant' concept as alternative, which results in classifiers of all sorts of plants, including evergreens and bloomers alike. Vice-versa, classifiers that match a more specific concept have more granularity; are less general with a higher accuracy, e.g., 'flower' now returns 'rose' and the results will always be a flower albeit a specific type of flower, i.e., roses and not a bracken, but nor another type of flower, e.g., a tulip. Concept expansion, hence, can use each edge in either way, downstream or upstream. In effect, when searching for a more granular concept, it selects the *target node* of more specific edges (downstream), and selects the *source node* of more abstract edges as well (upstream). For less granular concepts it takes the converse approach, flipping the downstream and upstream directions. Furthermore, when traversing more abstract edges subsequently, semantics will expand gradually by including more and more other categories of concepts. Vice versa, when traversing more specific edges subsequently, semantics will reduce by excluding categories of concepts that do not concur with the added details. Therefore, edge traversals in the unlimited semiosis subgraph decreases the semantic correspondence with the original query concept and should be avoided as long as possible. This coincides with a breadth-first approach, in which one single iteration over the knowledge base will address all children and all parents as alternative concepts first, before considering grandchildren and grandparents in the next iteration.

In our implementation, we use the following relations from ConceptNet: *IsA*, *hasSubEvent*, *PartOf* and *HasA*. The 'IsA' and 'PartOf' relations are directed towards more abstraction, whilst the 'HasSubEvent' and 'HasA' relations are directed towards the more specific concepts.

B. Paradigms

As explained in Section III.B, paradigms represent sets of disjoint concepts. When considering only relations in the external knowledge base that express paradigms, we consider the subgraph that emerges as a non-directed graph. In this *paradigmatic graph* we consider the query concept to represent the one and only connecting node of otherwise disconnected (undirected) graphs, each of them representing a paradigm. In other words, the paradigms for the query concept are constructed by performing a depth-first approach, each single branch from the query concept leading to another paradigm.

Application of *paradigms* provides us with the ability to reject alternatives, because that is the nature of paradigms: the user made a conscious choice for this query concept and therefore specifically *excludes* the paradigmatic alternatives.

That implies that for every classifier label that has a match with an alternative concept, that classifier is considered to be a paradigm of the query concept and hence is excluded from the results. For an additional application of paradigms, consider the alternative concepts that result from graph traversals from one of the other methods. These alternative concepts are checked whether they are paradigmatic to the query concept. If so, not only that alternative concept is rejected but the whole branch that is accessed through that concept is pruned from the search space. In this way, paradigms are applied with the aim to reduce the combinatorial explosion that occurs in the graph traversals of other methods. We currently conduct research into this additional application of paradigms – due to time and space restrictions we do not present the results of this experiment in this paper. In our ConceptNet example we consider *MemberOf* and *DerivedFrom* as paradigmatic relations.

C. Syntagms

The application of syntagms is not restricted to the semantic analysis. For instance, the use of the Stanford parser to decompose the natural language query into a structure of related query concepts represents an example of the use of the *syntagmatic* structure, applying linguistic rules. Another example is the translation of the query concepts (and subsequently their alternatives) into an instantiation of the meta-model. Here, the relations that exist between the entities in the meta-model (Figure 5) provide for the allowed syntagmatic combinations.

Application of syntagms to the semantic analysis relates to their power to facilitate transitions between realms of classifiers, as follows. Because each classifier is bound to only one entity in the meta-model, e.g., objects, by application of syntagms we can search for alternative concepts that belong to other entities of the meta-model, e.g., actions, or properties. In this way, we enable an otherwise ‘passive’ set of classifiers for alternative concepts. An example of this is the expansion of the action ‘person driving’ to objects such as ‘car’, ‘vehicle’, or ‘bike’. The knowledge that ‘driving’ relates to these objects is available in the knowledge base, and the syntagmatic relations reveal that knowledge. We conclude that the emerging *syntagmatic subgraph* is a directed, cyclic subgraph, in which edges represent transitions between entities in the meta-model.

In our ConceptNet example we consider the following relations to represent syntagms: *CapableOf*, *UsedFor*, *CreatedBy* reflect transitions from objects to actions; *Causes* reflect a transition from object to action; *hasProperty* from object to property.

VI. EXPERIMENT

In order to answer our research question *to what extent can semiotic structures contribute to the semantic interpretation of user queries?* we conducted an experiment.

In this experiment, we measure effectiveness and quality of different semiotic structures on the level of both semantic matching and image retrieval. The variable of the experiment is therefore represented by the differences in query expansion strategy, their core being the semiotic structures

that are explained in the previous section. The experiment context is defined by our TOSO dataset and 100 manually defined queries. More information on the TOSO dataset can be found in subsection A. The type of queries can be found in subsection B. The experiment variations and its baseline are explained in subsection C. The design of the experiment is presented in subsection D, and its evaluation is explained in subsection E.

A. Dataset

The TOSO dataset [9] consists of 145 images of toys and office supplies placed on a table top. In these images multiple objects can be present in several orientations as well as objects of the same type with different colors. In Figure 7, a sample of the dataset has been depicted. Examples of these objects are different types of cars, a bus, an airplane, a boat, a bus stop, a traffic light, different types of traffic signs, Barbies with different colored dresses, different colored plants, a water bottle, a screwdriver, a hamburger and a helmet. For this dataset, 40 relevant object classifiers, trained on table top images, are available as well as 11 attribute classifiers, which are colors. The object classifiers are trained with a recurrent deep convolutional neural network that uses a second stage classifier [6]. The colors are extracted using [39].



Figure 7. A sample of the TOSO dataset.

B. Queries

In this experiment, we created 100 queries. In the definition of the queries we used our prior knowledge of the available classifiers by intentionally choosing interesting expansions, for example, their synonyms or hypernyms. This was done by searching online thesauri, independently from our ConceptNet example. In this way, we created a set of queries that does not have direct matches to the available classifiers, but for which the use of semiotic structures could be helpful. These queries are divided into five equal groups based on their semiotic or semantic structure as follows:

- 1) **Synonym**: synonyms of our labels;
find the auto (classifier label: car);
- 2) **Unlimited semiosis**: hyponyms or hypernyms, i.e., parents or children of a label, or suspected ‘part of’ relations;
find the Mercedes (classifier label: car)
find the animal (classifier label: giraffe)
find the leaf (classifier label: plant)
- 3) **Paradigm**: excluding brothers and sisters in the graph (man vs. woman), restrictions to objects by color and/or spatial relations;

find the air vehicle (as opposed to land vehicle, e.g., car, bus, tram);

find the red sign on the right of the yellow car;

- 4) **Syntagm**: actions and properties related to our labels;

find the things landing (classifier label: airplane);

find the expensive things (classifier labels: airplane, car);

- 5) **Other**: words which have a less clear or vague relation with a classifier label:

find the flower pot (classifier label: plant)

find the traffic jam (classifier label: cars)

For each of the queries, we established a semantic ground truth as well as an image ground truth. The semantic ground truth was established by manually annotating for each classifier label in our classifier set whether it is irrelevant (0) or relevant (1) to the query. In our annotation, a classifier is *relevant* if (i) a classifier label is syntactically similar to a concept in the query, or (ii) a classifier label represents a synonym of a query concept. For the image ground truth we used the 145 test images from the TOSO dataset. An external annotator established the ground truth by annotating, for each query and for each image, whether the image was irrelevant (0) or relevant (1) to the query. Establishing relevancy was left to the annotator's judgement. For both the semantic and image annotations, the instructions indicated that all cases of doubt should be annotated as relevant (1).

C. Experimental variable

In the experiment, we compare the following query expansion methods, the implementation of which has been explained in Section IV:

- 1) SYNONYM (baseline)
- 2) UNLIMITED SEMIOSIS
- 3) PARADIGM
- 4) SYNTAGM
- 5) ALL

In the first method, which represents our baseline, we use the basic expansion over specific relations that are found in ConceptNet: *Synonym* and *DefinedAs*. In methods 2 (UNLIMITED SEMIOSIS), 3 (PARADIGM) and 4 (SYNTAGM), we apply query expansion by traversing only the relations that are particular to the subject semiotic structure (defined in Section V), however we added the relations from the baseline. In the ALL method (5) all possible relations from ConceptNet, excluding *TranslationOf* and *Antonym*, are applied for query expansion.

D. Experiment design

The design of the experiment is based on the hypothesis that a query will be served best by a query expansion strategy that shares its semiotic structures, e.g., the SYNTAGM expansion method will find most mappings for syntagm queries and perform worse for other queries. Therefore, in our experiment each expansion method from Section V will apply its one single expansion strategy over all query groups from Section VI.B; different methods will therefore perform differently, i.e., result in different mapping counts.

In order to test our hypothesis, we designed and ran two evaluation cases. The first evaluation case addresses the part

of the semantic gap that is about *semantic matching*. This case shows the impact of using semiotic structures on the effectiveness and quality of the mapping from the query to the classifier labels. The second evaluation case addresses the part of the semantic gap that is about *image retrieval*. This case shows the impact of semiotic structures on the effectiveness and quality of a full general-purpose image search engine.

E. Evaluation criteria

In our evaluations, we calculate the effectiveness and quality in terms of different types of F-measure for each query from Section VI.B. The following provides more detail for each evaluation case.

1) *Semantic Matching* In order to show the result of the expansion method on the mapping from the query to the classifier labels, we compare the result of each of the methods against the ground truth. This result is a list of classifier labels that are found by searching ConceptNet using the relations that are characteristic for the subject expansion method. In the evaluation we use two kind of metrics, corresponding to quality and effectiveness. The typical metric for quality is using precision, denoted P_{sg} , which takes into account the amount of true positives, i.e., found and annotated as relevant labels, and the total amount of found labels, i.e., true positives and false positives together, denoted as TP and FP, respectively:

$$P_{sg} = \frac{1}{n} * \sum_{q=1}^n \frac{TP_{sg}}{TP_{sg} + FP_{sg}}$$

where n denotes the total amount of queries.

The typical metric for measuring effectiveness is recall, denoted R_{sg} , which takes into account the amount of correctly found labels, i.e., true positives and the total amount of relevant labels, i.e., true positives and false negatives together, the latter denoted as FN:

$$R_{sg} = \frac{1}{n} * \sum_{q=1}^n \frac{TP_{sg}}{TP_{sg} + FN_{sg}}$$

where n denotes the total amount of queries.

Precision and recall are always an interplay, so we decided to not use precision and recall separately, but combine them by means of applying the F-measure. Since different applications can value the precision and recall of the semantic matching differently, the F_{β} -measure can be used to express that one should attach β times as much value to the recall results of the semantic matching than to its precision results. By using the F_{β} -measure as our primary means of evaluation, we can show the impact of the experiment results on three classes of applications, i.e., high quality applications that value precision over recall, high effectiveness applications that value recall over precision, and neutral applications that value precision equally important as recall. The F_{β} -measure is defined as:

$$F_{\beta} = (1 + \beta^2) * \frac{P_{sg} * R_{sg}}{(\beta^2 * P_{sg}) + R_{sg}}$$

For high quality applications, we put 10 times more emphasis on the precision and choose to use $\beta = 0.1$. For neutral applications we use the basic F-measure, i.e., $\beta = 1$ and for high effectiveness applications, we value recall 10 times more than precision and use $\beta = 10$. Naturally, these choices for β are made in order to show relative trends as opposed to an absolute judgement.

2) *Image Retrieval* The annotations are used in a similar way as on the level of the semantic matching. Again, F-score with $\beta = 0.1$ is used for high quality applications, $\beta = 1$ for neutral applications and $\beta = 10$ for high effectiveness applications.

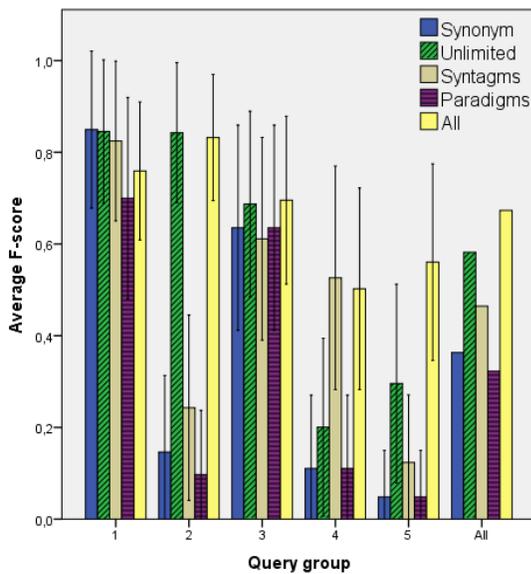
VII. RESULTS

In this section, we show the results of our experiment. The sections have the same structure as Section VI.D, so the first section explains the results about the semantic matching and the second section is about the results of the image retrieval.

For each of the evaluations, the assumption of normality was violated, as indicated by significant Kolmogorov-Smirnov statistics. We therefore present nonparametric Friedman-tests and Wilcoxon Signed-Ranks Tests to compare the different methods.

A. Semantic Matching

1) *High precision system ($\beta = 0.1$):* Graph 1 shows the F-score for the high precision system for each of the methods for each type of query group with the confidence interval of 95%. For two queries, both in group 4, no relevant annotation was available, so in group 4 analysis is done with 18 queries instead of 20 and in total 98 queries were analyzed.



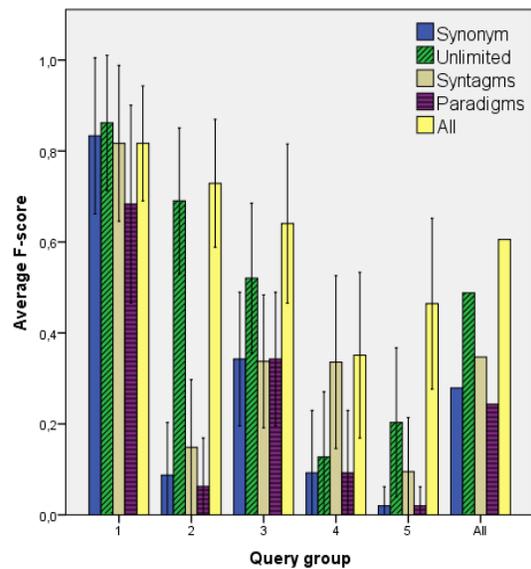
Graph 1. F-score Semantic Graph for High Quality.

A Friedman test showed a statistically significant difference among the methods ($\chi^2(4)=57.938$, $p<0.001$). Wilcoxon Signed-Ranks Test was used to follow up this finding. A Bonferroni correction was applied and all effects are reported at a 0.01 level of significance (0.05/5 conditions). The results can be found in Table I.

Table I. F-score All Semantic Graph Wilcoxon for High Quality.

	SYNO	UNL	SYNT	PARA	ALL
SYNO		Z=-4.458, p<0.001*	Z=-3.128, p=0.02*	Z=-1.890, p=0.059	Z=-5.224, p<0.001*
UNL	Z=-4.458, p<0.001*		Z=-2.401, p=0.016*	Z=-4.859, p<0.001*	Z=-2.162, p=0.031*
SYNT	Z=-3.128, p=0.02*	Z=-2.401, p=0.016*		Z=-3.635, p<0.001*	Z=-3.985, p<0.001*
PARA	Z=-1.890, p=0.059	Z=-4.859, p<0.001*	Z=-3.635, p<0.001*		Z=-5.635, p<0.001*
ALL	Z=-5.224, p<0.001*	Z=-2.162, p=0.031*	Z=-3.985, p<0.001*	Z=-5.635, p<0.001*	

The order of overall performance is thus SYNONYM = PARADIGM > SYNTAGM > UNLIMITED SEMIOSIS > ALL, all significant differences. For group 1 no significant differences between SYNONYM and the other methods are found. For group 2 significant differences between UNLIMITED SEMIOSIS and SYNONYM ($Z=-3.550$, $p<0.001$), SYNTAGM ($Z=-3.432$, $p=0.001$) and PARADIGM ($Z=-3.651$, $p<0.001$) are found. For group 3 no significant differences between PARADIGM and the other methods are found. For group 4 significant differences between SYNTAGM and SYNONYM ($Z=-2.670$, $p=0.008$) and PARADIGM ($Z=-2.670$, $p=0.008$) are found. For group 5 significant differences between ALL and SYNONYM ($Z=-3.053$, $p=0.002$), SYNTAGM ($Z=-2.833$, $p=0.005$) and PARADIGM ($Z=-3.053$, $p=0.002$) are found.



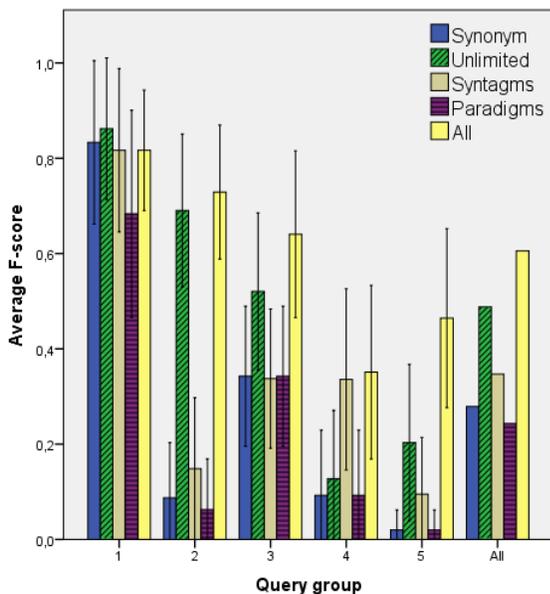
Graph 2. F-score Semantic Graph for Neutral.

2) *Neutral system ($\beta = 1$):* Graph 2 shows the F-score for the neutral system for each of the methods for each type of query group with the confidence interval of 95%. A Friedman test showed a statistically significant difference among the methods ($\chi^2(4)=98.571$, $p<0.001$). Wilcoxon Signed-Ranks Test was used to follow up this finding. A Bonferroni correction was applied and all effects are reported at a 0.01 level of significance (0.05/5 conditions). The results can be found in Table II.

Table II. F-score All Semantic Graph Wilcoxon for Neutral.

	SYNO	UNL	SYNT	PARA	ALL
SYNO		Z=-5.050, p<0.001*	Z=-3.043, p=0.02*	Z=-1.890, p=0.059	Z=-6.049, p<0.001*
UNL	Z=-5.050, p<0.001*		Z=-3.276, p=0.001*	Z=-5.366, p<0.001*	Z=-3.535, p<0.001*
SYNT	Z=-3.043, p=0.02*	Z=-3.276, p=0.001*		Z=-3.576, p<0.001*	Z=-5.329, p<0.001*
PARA	Z=-1.890, p=0.059	Z=-5.366, p<0.001*	Z=-3.576, p<0.001*		Z=-6.434, p<0.001*
ALL	Z=-6.049, p<0.001*	Z=-3.535, p<0.001*	Z=-5.329, p<0.001*	Z=-6.434, p<0.001*	

The order of overall performance is thus equal to the performance for the high quality system. The same significant differences are found for query group 1, 2, and 5. For group 3 significant differences between PARADIGM and UNLIMITED SEMIOSIS ($Z=-2.805$, $p=0.005$) and ALL ($Z=-3.237$, $p=0.001$) exist, as well as significant differences between UNLIMITED SEMIOSIS and SYNONYM ($Z=-2.805$, $p=0.005$), PARADIGM ($Z=-2.805$, $p=0.005$) and SYNTAGM ($Z=-2.926$, $p=0.003$). For group 4 an additional significant difference between SYNTAGM and UNLIMITED SEMIOSIS ($Z=-2.603$, $p=0.009$) is found.



Graph 3. F-score Semantic Graph for High Effectiveness.

3) *High effectiveness system ($\beta = 10$):* Graph 3 shows the F-score for the high effectiveness system for each of the methods for each type of query group with the confidence interval of 95%.

A Friedman test showed a statistically significant difference among the methods, $\chi^2(4)=108.197$, $p<0.001$. Wilcoxon Signed-Ranks Test was used to follow up this finding. A Bonferroni correction was applied and all effects are reported at a 0.01 level of significance (0.05/5 conditions). The results can be found in Table III.

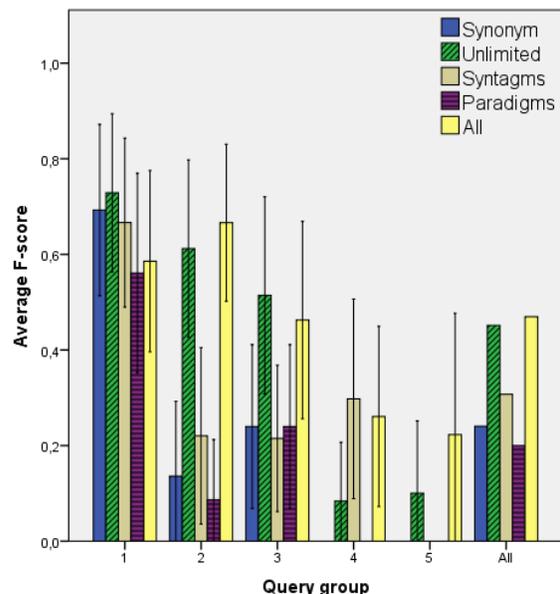
Table III. F-score All Semantic Graph Wilcoxon for High Effectiveness.

	SYNO	UNL	SYNT	PARA	ALL
SYNO		Z=-5.230, p<0.001*	Z=-3.241, p=0.001*	Z=-1.890, p=0.059	Z=-6.664, p<0.001*
UNL	Z=-5.230, p<0.001*		Z=-3.524, p<0.001*	Z=-5.521, p<0.001*	Z=-4.815, p<0.001*
SYNT	Z=-3.241, p=0.001*	Z=-3.524, p<0.001*		Z=-3.716, p<0.001*	Z=-6.083, p<0.001*
PARA	Z=-1.890, p=0.059	Z=-5.521, p<0.001*	Z=-3.716, p<0.001*		Z=-6.896, p<0.001*
ALL	Z=-6.664, p<0.001*	Z=-4.815, p<0.001*	Z=-6.083, p<0.001*	Z=-6.896, p<0.001*	

The order of overall performance is thus equal to the performance for both the high quality and neutral system. The same significance values are found as for the neutral system, except for the significant difference between PARADIGM and ALL in group 3. This difference is no longer significant for the high effectiveness system.

B. *Image Retrieval*

1) *High precision system ($\beta = 0.1$):* Graph 4 shows the F-score for the high quality system for each of the methods



Graph 4. F-score Image Retrieval for High Quality.

for each type of query group with the confidence interval of 95%. For 14 queries of which one in group 1, three in group 4 and ten in group 5, no relevant annotation was available. In total 86 queries are analyzed.

A Friedman test showed a statistically significant difference among the methods, $\chi^2(4)=58.891$, $p<0.001$. Wilcoxon Signed-Ranks Test was used to follow up this finding. A Bonferroni correction was applied and all effects are reported at a 0.01 level of significance (0.05/5 conditions). The results can be found in Table IV.

Table IV. F-score All Image Retrieval Wilcoxon for High Quality.

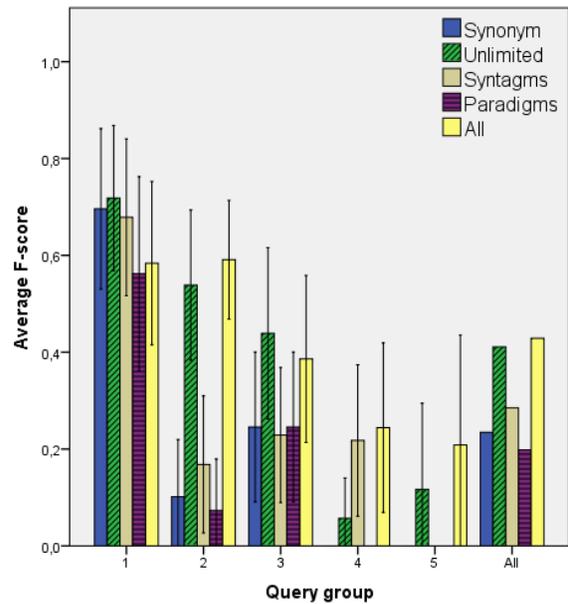
	SYNO	UNL	SYNT	PARA	ALL
SYNO		Z=-4.684, p<0.001*	Z=-2.511, P=0.012*	Z=-1.826, p=0.068	Z=-4.108, p<0.001*
UNL	Z=-4.684, p<0.001*		Z=-3.060, p=0.002*	Z=-5.012, p<0.001*	Z=-0.686, p=0.493
SYNT	Z=-2.511, p=0.012*	Z=-3.060, p=0.002*		Z=-3.155, p=0.002*	Z=-3.250, p=0.001*
PARA	Z=-1.826, p=0.068	Z=-5.012, p<0.001*	Z=-3.155, p=0.002*		Z=-4.722, p<0.001*
ALL	Z=-4.108, p<0.001*	Z=-0.686, p=0.493	Z=-3.250, p=0.001*	Z=-4.722, p<0.001*	

The order of overall performance is thus SYNONYM = PARADIGM > SYNTAGM > UNLIMITED SEMIOSIS = ALL, all significant differences. For group 1 no significant differences between SYNONYM and the other methods are found. For group 2 significant differences between UNLIMITED SEMIOSIS and SYNONYM ($Z=-3.294$, $p=0.001$), SYNTAGM ($Z=-2.982$, $p=0.003$) and PARADIGM ($Z=-3.413$, $p=0.001$) are found. For group 3 significant differences between PARADIGM and UNLIMITED SEMIOSIS ($Z=-2.701$, $p=0.007$) exist, as well as significant differences between UNLIMITED SEMIOSIS and SYNONYM ($Z=-2.701$, $p=0.007$), PARADIGM ($Z=-2.701$, $p=0.007$) and SYNTAGM ($Z=-2.845$, $p=0.004$). For group 4 no significant differences between SYNTAGM and the other methods are found and for group 5 no significant differences were found.

2) *Neutral system* ($\beta = 1$): Graph 5 shows the F-score for the neutral system for each of the methods for each type of query group with the confidence interval of 95%.

Table V. F-score All Image Retrieval Wilcoxon for Neutral.

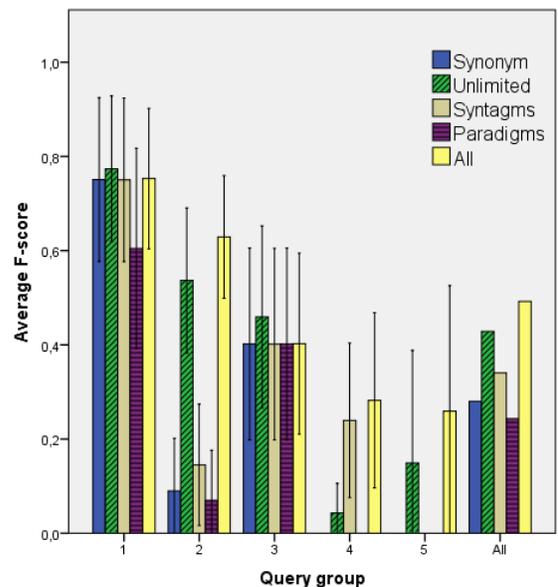
	SYNO	UNL	SYNT	PARA	ALL
SYNO		F=-4.723, p<0.001*	F=-2.354, p=0.019*	F=1.826, p=0.068	F=-4.019, p<0.001*
UNL	F=-4.723, p<0.001*		F=-3.350, p=0.001*	F=-5.045, p<0.001*	F=-0.800, p=0.424
SYNT	F=-2.354, p=0.019*	F=-3.350, p=0.001*		F=-3.052, p=0.002*	F=-3.554, p<0.001*
PARA	F=1.826, p=0.068	F=-5.045, p<0.001*	F=-3.052, p=0.002*		F=-4.704, p<0.001*
ALL	F=-4.019, p<0.001*	F=-0.800, p=0.424	F=-3.554, p<0.001*	F=-4.704, p<0.001*	



Graph 5. F-score Image Retrieval for Neutral.

A Friedman test showed a statistically significant difference among the methods, $\chi^2(4)=71.047$, $p<0.001$. Wilcoxon Signed-Ranks Test was used to follow up this finding. A Bonferroni correction was applied and all effects are reported at 0.01 level of significance (0.05/5 conditions). The results can be found in Table V. The same significant differences between conditions for ALL and the different query groups can be found as for the high quality system.

3) *High effectiveness system* ($\beta = 10$): Graph 6 shows the F-score for the high effectiveness system for each of the methods for each type of query group with the confidence interval of 95%.



Graph 6. F-score Image Retrieval for High Effectiveness.

A Friedman test showed a statistically significant difference among the methods, $\chi^2(4)=67.386$, $p<0.001$. Wilcoxon Signed-Ranks Test was used to follow up this finding. A Bonferroni correction was applied and all effects are reported at a 0.01 level of significance (0.05/5 conditions). The results can be found in Table VI.

Table VI. F-score All Image Retrieval Wilcoxon for High Effectiveness.

	SYNO	UNL	SYNT	PARA	ALL
SYNO		F=-4.586, p<0.001*	F=-2.825, p=0.005*	F=-1.826, p=0.068	F=-4.775, p<0.001*
UNL	F=-4.586, p<0.001*		F=-2.654, p=0.008*	F=-4.930, p<0.001*	Z=-2.257, p=0.024*
SYNT	F=-2.825, p=0.005*	F=-2.654, p=0.008*		Z=-3.362, p=0.001*	Z=-4.184, p<0.001*
PARA	F=-1.826, p=0.068	F=-4.930, p<0.001*	Z=-3.362, p=0.001*		Z=-5.196, p<0.001*
ALL	F=-4.775, p<0.001*	Z=-2.257, p=0.024*	Z=-4.184, p<0.001*	Z=-5.196, p<0.001*	

The same significant differences between conditions for ALL and the different query groups can be found as for the high quality and neutral system, except that we have no longer significant differences in group 3.

VIII. DISCUSSION

In the discussion, we reflect on the experimental results regarding the use of semiotic structures to close the semantic gap in CBIR applications, both its semantic matching and its image retrieval parts. Additionally, we discuss the limitations of this research.

A. Semantic matching

Results on the semantic matching show that the use of the ALL method for query expansion, e.g., taking into account each and every type of relation that is available in the knowledge base, gives best overall performance independent of the type of application you want to use, i.e., high quality, neutral or high effectiveness. This effect is mainly rooted in the *RelatedTo* relation; this relation does not reflect any semiotic structure and was hence excluded from the other methods, however, it leads to alternative concepts that appear relevant to the original query concept. Examples of such relations produce expansions such as 'key fob' to 'key ring' and 'aircraft' to 'airplane', which can fuel a debate whether their relation with the query concept would not be better expressed as *synonym*. The method UNLIMITED SEMIOSIS gives the second best overall performance for all types of applications, both significantly lower than ALL and significantly higher than the other methods. As expected, this method turns the external knowledge base into a directed graph that expresses levels of aggregation, and therefore finds more abstract or more specific concepts compared to the baseline. The method SYNTAGMS has the third overall performance. This semiotic type is particularly good due to the fourth group of queries where a translation to another syntagmatic type is due for success, i.e., from *action* to

object or from *attribute* to *object*. PARADIGMS have equal performance, or even slightly worse, than BASELINE. Whereas our hypothesis was that it would only exclude irrelevant concepts, it also excluded some concepts that were annotated as relevant, such as 'soccer_ball' for 'ball' and 'motorbike' for 'motorcycle'. Whether these relations should be excluded or included is based on the type of application that it is used for. Alternatives that are found by this semiotic type, and that are causing a decrease in performance, are all found as *synonym* as well. Hence, a strategy might be to include all relations that have the synonym relation independent of the presence of the PARADIGMS.

A closer look at the results for the different query groups as introduced in Section VI.B shows the following. For the first query group (*baseline*) no significant differences are found over the different expansion strategies. Surprisingly, no SYNONYM relation between 'key fob' and 'keyring', and between 'aircraft' and 'airplane' is available in ConceptNet. The first is present through a *RelatedTo* relation and the second through an *IsA* relation. Furthermore, no ConceptNet entry for 'camping bus' is present, so no expansions are found, dropping performance. Performance for PARADIGMS is lowest, because expansions for 'automobile' and 'beefburger', which are both considered relevant according to our ground truth, are paradigmatically excluded, dropping performance. SYNTAGMS is slightly lower than SYNONYM, because of an, as irrelevantly annotated, relation between 'shit' and 'cow', which is a debatable choice. UNLIMITED SEMIOSIS finds a relation between 'football' and 'skateboard', which slightly decreases performance. ALL has found several good alternatives, but also irrelevant ones, which is nicely visible in Graph 1 (relatively low F-score) and Graph 3 (relatively high F-score).

For the second query group (*unlimited semiosis*) the hypothesis was that UNLIMITED SEMIOSIS performs best. Significant differences between all other expansion methods, except ALL, are found. As the different graphs show, UNLIMITED SEMIOSIS has a better quality (Graph 1), whereas ALL has a higher effectiveness (Graph 2). The ALL method finds additional relevant concepts for 'animal' ('cow'), 'tool' ('screwdriver') and 'door' ('bus'), but irrelevant concepts for 'vehicle' ('tag') and 'leaf' ('pig'). The other methods find very little concepts and, therefore, performance is low.

The third query group (*paradigms*) is a group that expresses restrictions, such as spatial relation and color. No significant differences in performance are found for high precision applications, but for neutral and high effectiveness applications performance of UNLIMITED SEMIOSIS and ALL methods are significantly higher than PARADIGMS. This is due to relevant expansions for 'animal', 'flower', 'vehicle', 'hat', 'Mercedes' and 'Range Rover'. No results for 'air vehicle', 'water vehicle' and 'land vehicle' are found.

For the fourth query group (*syntagms*) the hypothesis was that the SYNTAGMS method performs best. As with the second query group and the UNLIMITED SEMIOSIS method, we see a high-quality for the SYNTAGMS method, but a high recall for the ALL method. The other methods have low performance, because they remain in the same syntagmatic part of the graph, whereas this query group requires a

transition to other syntagmatic alternatives. The main difference between SYNTAGMS and ALL is rooted in 'riding', 'stopping' and 'fast' in favor of ALL and 'landing' in favor of SYNTAGMS.

Finally, the fifth query group (*others*) are queries that have a very loose relation with the classifiers. Results show a significant difference between the ALL method and the other methods, as expected, but not compared to the UNLIMITED SEMIOSIS method. Many concepts in this group can only be found by ALL, but for 'headgear', 'tomato', 'farm' and 'wool', concepts are also found by the UNLIMITED SEMIOSIS method.

In the context of this case of the experiment, we can conclude that the type of query and the type of application prescribe the type of semiotic methods to consider. For applications that value effectiveness, the ALL method will be a good choice. Contrarily, for applications that require high quality, the UNLIMITED SEMIOSIS method would be a better choice, assuming that its queries do not require transitions between syntagmatic concepts (group 4), or are vaguely related to classifiers (group 5). Another good option for high quality applications would be to combine the SYNTAGMS and UNLIMITED SEMIOSIS methods. Finally, although in theory the PARADIGMS method should improve results for high quality applications, results indicate that it needs a more careful approach.

B. Image retrieval

Results on the image retrieval case show the impact of the semiotic structures on both parts of the semantic gap, and therefore the system as a whole. The general trend is that performance for this case is lower than for the semantic matching case, above. This originates from the fact that our classifiers do not perform very well. For instance, in query groups 4 (*syntagms*) and 5 (*other*) some expansion methods show no performance at all, which implies that despite the presence of relevant ground truth for them, none of the queries produce image results. The largest difference in overall performance between both cases is that the methods for UNLIMITED SEMIOSIS and ALL are no longer significantly different (Graphs 4 – 6). This is an indication that by adding irrelevant concepts (by the ALL method) more irrelevant images are produced, which might hurt more than adding less relevant concepts (by the UNLIMITED SEMIOSIS method) that produces less irrelevant images. This result even holds for high effectiveness applications.

A closer look at the results for the different query groups as introduced in Section VI.B shows the following. For the first group (*Synonyms*) not much difference is found over the various methods. Only the ALL methods drops a little more than the SYNTAGMS method, because the expansion by the ALL method from 'motorbike' to concepts 'horse' and 'helmet' really hurts performance as both are not synonyms while any image with either a horse, a helmet or a motorcycle will still be retrieved. As indicated above, the performance of the image retrieval case is lower than the semantic matching case. In this query group that is exemplified by the fact that although our classifiers for 'boat', 'motorcycle' and 'turd' are performing flawless,

'car', 'bus', 'traffic light' and 'turnscrew' perform less optimal ($F_{0.1} \sim 80\%$), whilst the classifiers for 'airplane', 'helmet' and 'football' can at best be graded acceptable ($F_{0.1} \sim 60\%$).

In the second query group (*unlimited semiosis*), the UNLIMITED SEMIOSIS method performs best. Significant differences between all other expansion methods, except ALL, are found. Differently from the results in the Semantic Matching, the UNLIMITED SEMIOSIS method is not better than ALL for high quality applications.

Results from the third query group (*paradigms*) interestingly show that the UNLIMITED SEMIOSIS method is slightly, but not significantly, better than its counterpart ALL, even for high recall applications. However, this is not only because of irrelevant expansions by the ALL method. In this group, many paradigmatic restrictions are specified by the queries, specifically about color, and colors cause a large decrease in performance in image retrieval. For example, the ALL method produces a semantic match between 'silver' and 'gray', indicating that gray cars are relevant. Unfortunately, in the image retrieval part silver cars are not detected as silver, but mainly as black. This is because many of the cars have black windows. Another example shows that green traffic lights are never detected, because the main color of the traffic light is black, irrespective of the light that is lit. In fact, this represents a typical example for unlimited semiosis where the semantic value of 'green' refers to an abstraction level that is far above the specific level that is indicated by the minimal part of the object that actually represents the green lit light. After all, we are not searching for a completely green traffic light. Besides the color classifiers, also other classifiers perform suboptimal, which has a negative effect on the results. Because, when a classifier is not able to detect the relevant concept in a relevant image, no difference between the methods can be registered.

Image retrieval results in the fourth group (*syntagms*) show similar results as in the semantic matching case: a slightly higher quality for the SYNTAGMS method and a higher effectiveness for the ALL method. These differences are, however, not significant any more. This is also the case for the fifth query group (*others*): no differences compared to semantic graph results, while the results are not significant anymore.

Overall, we can thus conclude that for high quality applications, the ALL method potentially hurts performance. Already for neutral applications, the UNLIMITED SEMIOSIS method, or a combined application of the SYNTAGMS and UNLIMITED SEMIOSIS methods might be a better choice than the ALL method. Additionally, this conclusion might prove stronger when taking into account the end user of the system whom might judge the results from the ALL method far worse than the results from the semiotic methods: in a retrieval system with many irrelevant results, as with application of the ALL method, it would be hard to find the relevant results amongst them, whilst the less, but more relevant results of the UNLIMITED SEMIOSIS method will be much easier to detect by the end user.

C. Limitations of experiment

One of the limitations of these experiments is that our dataset is really small. With only 51 classifiers, the probability that any of the words in ConceptNet matches our classifier labels is, therefore, much lower. Then, one single true positive has a major impact on score whilst the many false positives that happen to have no match do not add to the score balance. This might be the reason that the ALL method is performing better than we expected.

A second limitation is performance of the classifiers. As explained in part B. of this section, our color classifiers as well as some object classifiers are suboptimal. In order to profit from improvements in the semantic reasoning part of the system, good classifiers are needed. This argument also holds in reverse: on optimizing classifiers, overall little will be gained unless the improvements in this part of the semantic gap is matched with an equal improvement in the semantic matching part of the semantic gap.

An algorithm performs only as good as the quality of the data it is provided with. Especially when the focus is on generic semantic knowledge, a third limitation is the knowledge base of choice. ConceptNet has a lot of different types of relations and, therefore, connections between concepts exists that apply different relations than expected, i.e., impacting accuracy, or no relations are available at all where one would expect their occurrence, impacting completeness. Although we experienced major improvements of version 5.3 over 5.2, e.g., corrections from erroneous relationships, several flaws in our experiment find their root in debatable concept relations from ConceptNet, or absent concepts. Another lesson learned from ConceptNet is the use of underscored words. Underscored words represent complex concepts that are represented by composition of two or more words by applying underscores, e.g., 'woman_wardrobe' or 'red_traffic_light'. Humans easily recognize their (syntagmatic) structure, but putting such understanding into (semiotic) rules is another matter completely. Therefore, we decided to abandon their use altogether, in order to stay away from potentially incorrect expansion results from factually correct data such as *CapableOf(camper, shoe_away_bear)* and *PartOf(dress, woman_wardrobe)*.

Finally, we have designed the experiment to score against two ground truths, one for the semantic matching and one for the image retrieval. They have the 100 queries in common, and since we have only 20 queries for each query group (Section VI.B) they also share their susceptibility to annotation-induced performance variations. We acknowledge this weakness in our experiment, especially since each annotation is performed by one individual each.

IX. CONCLUSION AND FUTURE WORK

In conclusion, applying semiotic relations in query expansion over an external, generic knowledge base, contributes to a higher quality semantic match between query concepts and classifier labels, and also significantly improves image retrieval performance compared to a baseline with only synonym expansions. The type of query

and the type of application prescribe the type of semiotic methods that should be considered for semantic matching. The indiscriminate use of all available relations that are present in the external knowledge base potentially hurts performance of the image retrieval part. The same approach for the semantic matching surprisingly outperformed the dedicated semiotic methods, although we have strong reasons to believe this effect is rooted in coincidental flaws in the knowledge base of choice. The experiment results also confirmed that the semantic gap that is experienced within CBIR consists of two cascading parts, and that little is gained overall when improvements address one part only. Finally, although multiple relations from the external knowledge base have been mapped onto one single semiotic method that at best approximates the semantics of the underlying relations, it is above doubt that semiotic coherence emerges in the otherwise non-semiotic semantic network that the external knowledge base represents. We have shown that this semiotic coherence can be employed to improve the semantic capability of a software system.

In future research, it is advisable to explore the effectiveness of these semiotic structures on other knowledge bases, containing either generic or domain-specific knowledge, in order to further evaluate the true genericity of this semiotic approach. Specifically related to ConceptNet it may be worthwhile to investigate appropriate (semiotic) ways to handle complex concepts (underscored words) in order to disclose their knowledge and improve query expansion.

Inclusion of more classifiers, including better color classifiers, and more classifier types, such as action classifiers and object relation classifiers, will improve the significance of the outcome of the experiments as well as the applicability of the expansion methods.

Furthermore, it would be interesting to conduct research into the influence of other semiotic structures, such as the semiotic square about contradictions, expressing relations that are also available in external databases, e.g., negated concepts and antonyms.

Additionally, it would be beneficial to measure image retrieval performance using relevance feedback from an end user on the found classifier labels by ConceptNet. For instance, our use of paradigms is completely unaware of the intentions of the end user and therefore might wrongly exclude a specific set of paradigmatic concepts. This can be easily corrected by adding context of use through relevance feedback.

ACKNOWLEDGMENT

This research was performed in the context of the GOOSE project, which is jointly funded by the enabling technology program Adaptive Multi Sensor Networks (AMSN) and the MIST research program of the Dutch Ministry of Defense. Furthermore, we acknowledge the ERP Making Sense of Big Data for their financial support. The authors want to express their gratitude to Dignée Brandt for her annotation of the ground truth, and to Charelle Bottenheft for her support to the statistical analysis.

REFERENCES

- [1] M. H. T. de Boer, L. Daniele, P. Brandt, and M. Sappelli, "Applying Semantic Reasoning in Image Retrieval," in *ALLDATA 2015, The 1st Int. Conf. on Big Data, Small Data, Linked Data and Open Data*, 2015, pp. 69–74.
- [2] K. Schutte et al., "Interactive detection of incrementally learned concepts in images with ranking and semantic query interpretation," in *CBMI*, 2015, pp. 1–4.
- [3] K. Schutte et al., "GOOSE: Semantic search on internet connected sensors," in *SPIE Defense, Security, and Sensing*, 2013, p. 875806.
- [4] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 12, pp. 1349–1380, 2000.
- [5] P. G. B. Enser, C. J. Sandom, J. S. Hare, and P. H. Lewis, "Facing the reality of semantic image retrieval," *J. Doc.*, vol. 63, no. 4, pp. 465–481, 2007.
- [6] H. Bouma, P. T. Eendenbak, K. Schutte, G. Azzopardi, and G. J. Burghouts, "Incremental concept learning with few training examples and hierarchical classification," *Proc. SPIE*, vol. 9652, 2015.
- [7] J. Schavemaker, M. Spitters, G. Koot, and M. de Boer, "Fast re-ranking of visual search results by example selection," in *CAIP 2015, Part I, LNCS 9256*, 2015, pp. 387–398.
- [8] R. Speer and C. Havasi, "Representing General Relational Knowledge in {ConceptNet} 5.," in *LREC*, 2012, pp. 3679–3686.
- [9] K. Schutte et al., "TOSO dataset." TNO, The Hague, The Netherlands, 2015.
- [10] M. H. T. de Boer et al., "Ground Truth on 100 User Queries about TOSO Dataset," 2015. [Online]. Available: DOI:10.13140/RG.2.1.3688.9049.
- [11] J. Deng et al., "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
- [12] P. Perona, "Vision of a Visipedia," *Proc. IEEE*, vol. 98, no. 8, pp. 1526–1534, 2010.
- [13] L. Bai, S. Lao, G. J. F. Jones, and A. F. Smeaton, "Video semantic content analysis based on ontology," in *IMVIP 2007*, 2007, pp. 117–124.
- [14] A. D. Bagdanov, M. Bertini, A. Del Bimbo, G. Serra, and C. Torniai, "Semantic annotation and retrieval of video events using multimedia ontologies," in *ICSC 2007*, 2007, pp. 713–720.
- [15] A. Oltramari and C. Lebiere, "Using ontologies in a cognitive-grounded system: automatic action recognition in video surveillance," in *Proc. Int. Conf. STIDS*, 2012.
- [16] G. Erozal, N. K. Cicekli, and I. Cicekli, "Natural language querying for video databases," *Inf. Sci. (Ny)*, no. 178, pp. 2534–2552, Apr. 2008.
- [17] X. Chen, A. Shrivastava, and A. Gupta, "NEIL: Extracting visual knowledge from web data," in *CICCV 2013*, 2013, pp. 1409–1416.
- [18] M. de Boer, K. Schutte, and W. Kraaij, "Knowledge based query expansion in complex multimedia event detection," *Multimed. Tools Appl.*, pp. 1–19, 2015.
- [19] D. D. K. Sleator and D. Temperley, "Parsing English with a link grammar," *arXiv Prepr. C.*, 1995.
- [20] M. Nazrul Islam, "A systematic literature review of semiotics perception in user interfaces," *J. Syst. Inf. Technol.*, vol. 15, no. 1, pp. 45–77, Mar. 2013.
- [21] D. Lamas and H.-L. Pender, "Reflection on the role of semiotic engineering in co-design of interaction," *IEEE Lat. Am. Trans.*, vol. 12, no. 1, pp. 48–53, Jan. 2014.
- [22] F. Nicastro et al., "A semiotic-informed approach to interface guidelines for mobile applications: A case study on phenology data acquisition," in *ICEIS 2015*, 2015, vol. 3, pp. 34–43.
- [23] B. Shishkov, J. L. G. Dietz, and K. Liu, "Bridging the language-action perspective and organizational semiotics in SDBC," in *ICEIS 2006*, 2006, pp. 52–60.
- [24] C. M. de A. Barbosa, R. O. Prates, and C. S. de Souza, "MARq-G*," in *CLHC '05*, 2005, vol. 124, pp. 128–138.
- [25] A. L. O. Paraense, R. R. Gudwin, and R. de Almeida Goncalves, "Brainmerge: a Semiotic-Oriented Software Development Process for Intelligence Augmentation Systems," in *2007 Int. Conf. on Integration of Knowledge Intensive Multi-Agent Systems*, 2007, pp. 261–266.
- [26] B. Shishkov, M. van Sinderen, and B. Tekinerdogan, "Model-Driven Specification of Software Services," in *ICEBE '07*, 2007, pp. 13–21.
- [27] C. Landauer, "Layers of Languages for Self-Modeling Systems," in *IEEE SASO 2011*, 2011, pp. 214–215.
- [28] M. Kwiatkowska, K. Michalik, and K. Kielan, *Soft Computing in Humanities and Social Sciences*, vol. 273. Springer Berlin Heidelberg, 2012.
- [29] S. Yi'an, "A survey of peirce semiotics ontology for artificial intelligence and a nested graphic model for knowledge representation," in *CEUR Workshop Proc.*, 2013, vol. 1126, pp. 9–18.
- [30] S. Chai-Arayalert and K. Nakata, "Towards a semiotic approach to practice-oriented knowledge transfer," in *Proc. KMIS 2012*, 2012, pp. 119–124.
- [31] S. Liu, W. Li, and K. Liu, "Pragmatic Oriented Data Interoperability for Smart Healthcare Information Systems," in *2014 14th IEEE/ACM Int.Symp. on Cluster, Cloud and Grid Computing*, 2014, pp. 811–818.
- [32] J. Yoon, "Improving recall of browsing sets in image retrieval from a semiotics perspective," Ph.D. dissertation, University of North Texas, 2006.
- [33] E. Hartley, "Bound together: Signs and features in multimedia content representation," in *Proc. COSIGN 2004*, 2004.
- [34] D. Chandler, *Semiotics: the basics*, 2nd ed. Routledge, 2007.
- [35] C. S. Peirce, *Collected Papers of Charles Sanders Peirce: Elements of logic*, vol. 2. Cambridge, MA: Harvard University Press, 1932.
- [36] U. Eco, *A theory of semiotics*. Bloomington, IN: Indiana University Press / London: Macmillan, 1976.
- [37] D. Silverman and B. Torode, *The Material Word: Some Theories of Language and its Limits*, In: London: Routledge & Kegan Paul, 1980.
- [38] M.-C. De Marneffe, B. MacCartney, C. D. Manning, and others, "Generating typed dependency parses from phrase structure parses," in *Proc. LREC*, 2006, vol. 6, no. 2006, pp. 449–454.
- [39] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, 2009.

Recognition of Human Faces in the Presence of Incomplete Information

Soodeh Nikan and Majid Ahmadi

Electrical and Computer Engineering

University of Windsor

Windsor, ON, Canada

nikan@uwindsor.ca and ahmadi@uwindsor.ca

Abstract—Proposed face recognition in this paper is a block based approach. Gabor magnitude-phase centrally symmetric local binary pattern has been used to extract directional texture characteristics of face at different spatial frequencies. Centrally symmetric local binary pattern is applied on the sub-blocks of magnitude and phase responses of Gabor images, which is the important contribution of the proposed work. Sparse classifier is employed as local classifier to find the sub-blocks class labels. We have evaluated the performance of the proposed algorithm on AR and ORL databases. In real world scenarios, partial face images are available to recognize the identity of an unknown individual. By comparing the recognition accuracy on the recognition results of image sub-blocks, we find the location and size of the most effective face sub-region for identification. Moreover, Chi-Square weighted fusion of image sub blocks at decision level leads to significantly improved recognition accuracy. We also evaluate the performance of the proposed algorithm in the presence of incomplete information for low resolution and occluded images.

Keywords—face recognition; block based; effective subregion; partial image; incomplete information.

I. INTRODUCTION

Face recognition is widely used as a biological identification technique which is applied to recognize an unknown individual by analyzing and comparing their facial image to the available database of known identities. It has a wide range of applications such as social networking, access control, forensic images, surveillance cameras, and law enforcement [1]. The accuracy of face recognition is affected by challenging conditions due to partial occlusion, low resolution, poor illumination, head pose variation, facial expression and blur effect. In recent years, many identification techniques were proposed in order to increase the accuracy of face recognition versus degrading conditions [1]. In holistic based approaches the whole face area is employed to extract features and deciding on the identity label. A robust image representation against occlusion and illumination variation was proposed in [2] using the combination of subspace learning and cosine-based correlation approach, which was applied on the orientation of gradient. However, local based techniques by dividing image into sub-regions and fusion of the extracted features or classification results, leads to robustness against variations in the appearance. Local Gabor binary pattern histogram (LGBPH) technique was proposed in [3], where the local binary pattern (LBP) histograms of sub-blocks of Gabor magnitude images were combined. Different

sub-blocks were differentiated in concatenation of features, by assigning a Kullback–Leibler divergence (KLD) weight to the corresponding sub-blocks. In [4] a block-based face recognition technique was proposed by extracting uniform LBP histograms. The results of local nearest neighbour classifiers were combined using an entropy weighted decision fusion to reduce the effect of sub-blocks with less information content. Local phase quantization (LPQ) and multi-scale LBP were applied on the proposed gradient based illumination insensitive representation of image sub-blocks in [5]. Weighted fusion at score and decision level found the identity of unknown individuals. In [6] the gray values of pixels in image sub-regions were concatenated and class specific multi sub-region based correlation filter bank technique (MS-CFB) was calculated for the training samples and test images. Local polynomial approximation (LPA) filter and directional scale optimization was proposed in [7]. LBP directional images were divided into sub-blocks at four levels. Finally, linear discriminant analysis (LDA) was applied on the concatenation of local histograms at four levels. Nevertheless, some facial areas that contain non-discriminative information can be excluded in the recognition process and computational complexity is reduced by analyzing fewer image sub-blocks instead of the whole face area [1]. We need to find the most effective sub-image to identify an unknown individual. This technique is very effective when some parts of the face are occluded by an external object. In some application, such as images acquired by surveillance cameras or forensic images, we have incomplete information from low resolution or partial face images. In some cases only a partial image of the face with a small amount of discriminative information is available. The proposed approach in [8] addressed partial face recognition using an alignment-free combination of multi-keypoint descriptors (MKD) and sparse representation-based classification (SRC). A set of MKDs were applied on images in the gallery set and a partial probe image was represented as a sparse linear combination of gallery dictionary.

This paper is built upon our proposed work in [1]. However, in this work, we proposed a weighted fusion technique to differentiate the effect of image sub regions on the identification decision based on their discriminative effectiveness. The image is divided into sub-blocks and the proposed face recognition technique, which is shown in Fig. 1, is applied on local areas. The size and location of the most effective area of the face in identification process has been investigated through the experiments on four different databases. We proposed Gabor magnitude-phase centrally

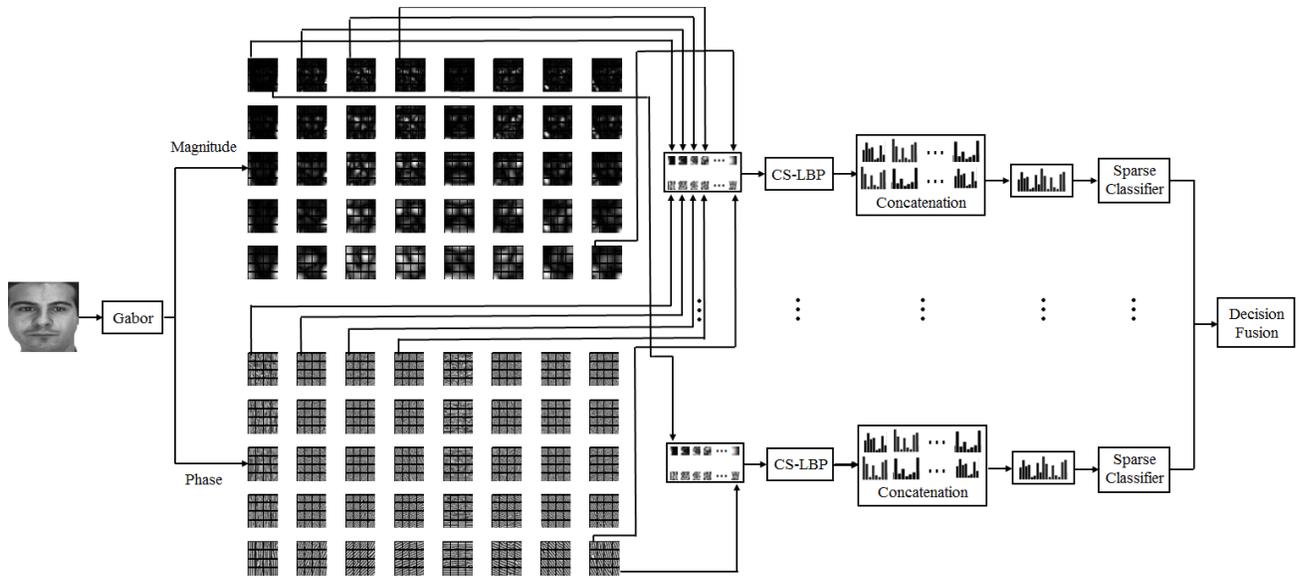


Figure 1. Block diagram of the proposed face recognition technique [1].

symmetric local binary pattern (GMP-CS-LBP) technique as feature extractor based on the symmetry in a local area around image pixels [1, 9]. In order to include the magnitude and phase information of the local characteristics of face, which are insensitive against appearance changes, we have applied texture descriptor on the magnitude and phase responses of Gabor images. The extracted features are concatenated for each image sub-block. Sparse classifier is employed on image sub-regions to find the local class labels [1]. In this paper, we propose a weighted majority voting (MV) scheme, which combines local decisions. The Chi-Square (CSQ) distance measurement [10] of the histograms of image sub blocks is applied as the local weights. We also evaluate the performance of the proposed technique in the presence of incomplete information due to low resolution and partial occlusion.

The rest of paper is organized as follows. In Section II, the configuration of feature extraction technique is explained in detail. Section III describes the classification approach. Section IV provides the experimental results. The paper is concluded in Section V.

II. FEATURE EXTRACTION

The proposed GMP-CS-LBP feature extraction in this paper is the fusion of magnitude and phase information of Gabor coefficients. Configuration of the proposed feature extraction technique is shown in Fig. 1 [1].

A. Gabor Filter

Gabor filter extracts the characteristics of signal at different scales and orientations, which resembles the mammalia response of vision cells. In order to acquire directionally selective local properties of a face image at various spatial frequencies, which are invariant against appearance changes due to expression and illumination variations, 2-D Gabor filters at S_{max} scales and O_{max} orientations are convolved by image [1]. Gabor filters are

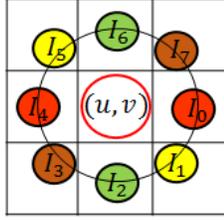
obtained as follows by ranging the spatial scale s from 1 to S_{max} and orientation o from 1 to O_{max} [11, 12],

$$\psi_{s,o}(x,y) = \frac{q_{s,o}^2}{\sigma^2} \cdot e^{-\left(\frac{z^2 q_{s,o}^2}{2\sigma^2}\right)} \cdot \left[e^{(jzq_{s,o})} - e^{-\left(\frac{\sigma^2}{2}\right)} \right], \quad (1)$$

where $q_{s,o} = q_s \exp(j\theta_o) = [\pi/2(\sqrt{2})^s] \exp(j\pi o/8)$ (in this paper, we defined 5 scales and 8 orientations). $z = (x, y)$, and $\sigma = 2\pi$ [11, 12]. The magnitude and phase responses of Gabor filtered image are shown in Fig. 1 [1].

B. Centrally Symmetric Local Binary Pattern (CS-LBP)

One of the most powerful local descriptors where the texture information are analysed by comparing the intensity value of local texture in a small neighbourhood and suppress the monotonic offset of neighbour pixels is local binary pattern (LBP) analysis. LBP is a very fast technique and easy to execute [9, 12]. In a circular neighbourhood with radius R and P neighbours around each image pixel, we compared the neighbours with the centre pixel and depending on the sign of their difference a 1 or 0 value (for positive difference or negative difference, respectively) is assigned to the corresponding neighbours. Therefore, a P-bit binary pattern is associated with the centre pixel. Thus, for image pixels we have decimal values ranging from 0 to 2^P , which are used to construct a histogram of 2^P -bin as the texture features. We can reduce the number of histogram bins, which decreases the size of extracted features by employing the symmetry in the local area around each pixel. In centrally symmetric LBP (CS-LBP) technique [9], the centre symmetric pairs of neighbours are compared instead of comparing each of them with the centre, as shown in Fig. 2. Therefore, the range of decimal values is reduced to $0 - 2^{(P/2)}$ and the stability of the extracted features



$$CS_{\text{pattern}}(u, v) = \frac{\{F(I_0 - I_4).2^0 + F(I_1 - I_5).2^1 + F(I_2 - I_6).2^2 + F(I_3 - I_7).2^3\}}{F(I_2 - I_6).2^2 + F(I_3 - I_7).2^3}$$

Figure 2. Calculation of CS-LBP for a pixel at (u, v) [1].

against flat texture is increased. The calculation of decimal value associated with the binary patterns is as follows [1, 9],

$$CSLBP_{\text{dec}}(u, v) = \sum_{l=0}^{(P/2)-1} F(I_l - I_{l+(P/2)}) 2^l,$$

$$\text{where } F(x) = \begin{cases} 1 & x \geq Th. \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

(u, v) is the position of centre pixel and I_l is the intensity value of l^{th} neighbor of the centre. R and P are 1 and 8 in this paper. In order to increase the stability against flat areas, the intensity differences between centre symmetric pairs are compared to a threshold value (Th) greater than 0, which is used as threshold in LBP technique [9]. The value that is assigned as threshold is defined in the following section [1].

C. Local GMP-CS-LBP Histograms

In order to employ magnitude and phase information simultaneously, CS-LBP technique is applied on the magnitude and phase responses of Gabor images at different scales and orientations. However, the threshold value in (2) is different for comparing magnitude or phase information. Through the exhaustive search, in this paper we employ 0.1 as the magnitude threshold and 90° as phase threshold. Following by calculation of the binary patterns and the corresponding decimal values of image pixels and constructing histograms, the $2^{(P/2)}$ -bin magnitude and phase histograms are concatenated [1].

Furthermore, to find the most effective sub region of face image on the identification accuracy, we divide Gabor images into rectangular non overlapping sub blocks of $m \times n$ pixels. By concatenating the histograms of magnitude and phase responses of all scales and orientations of Gabor responses, we obtain a histogram of $2^{(P/2)+1} \times S_{\text{max}} \times O_{\text{max}}$ bins for each image sub region [1].

III. SPARSE CLASSIFICATION

Local classifiers are based on the sparsest representation of the probe sample using the combination of corresponding gallery samples of the same class label [13]. Image samples, which are belonging to the same individual, lie on a linear subspace [1].

$$g = [g_1, g_2, g_3, \dots, g_M]. \quad (3)$$

$$g_i = [f_1^g, f_2^g, f_3^g, \dots, f_N^g]. \quad (4)$$

Where g is gallery dictionary, which is including all gallery samples in the database. g_k is matrix of k^{th} class of subject, which consists of gallery feature vectors as its columns (f_k^g is the feature vector of the k^{th} sample in g_k), where M and N are the number of classes and gallery samples per class, respectively. Therefore, using the matrix of gallery dictionary and a coefficient vector we can define the feature vector of a probe sample as a linear combination as follows [1, 13],

$$f_i^p = g.B. \quad (5)$$

Where $B = [0, 0, \dots, 0, \beta_1^k, \beta_2^k, \dots, \beta_N^k, 0, 0, \dots, 0]$ and β_j^k is the j^{th} coefficient corresponding to the k^{th} class. The sparsest representation of probe sample can be achieved, if only the coefficients associated with class label of the probe sample are non-zero. Those coefficients are calculated using the l_1 -norm solution of equation (5) and the identity label of the probe sample as follows [1, 13].

$$(l_1): \hat{B}_1 = \text{argmin} \|B\|_1 \quad \text{while } f^p = g.B. \quad (6)$$

IV. DECISION FUSION

In order to combine the local result on the image sub blocks and come up with a final decision on the identity of the unknown probe sample, majority voting scheme is applied as the decision fusion strategy. The votes of the image sub blocks are combined by adding up the local votes for each class of subject. Finally, the class of identity with maximum total votes is selected as the final decision [1].

V. LOCAL WEIGHTING

In order to differentiate between the effects of image sub blocks on the final decision on the identity of the probe sample, local weights are calculated on image sub regions. In this paper, we calculate the Chi-Square (CSQ) distance [10] between the histograms of probe image sub blocks and the local histograms in the class-prototype of corresponding class of probe sub-block. The class-prototype (CP) image in the proposed approach in this paper is the average image of all gallery samples belonging to each class of subject as follows.

$$W_{CS}(PI^m, CP^m) = \sum_{j=1}^{N_b} \frac{(PI_{b_j}^m - CP_{b_j}^m)^2}{PI_{b_j}^m + CP_{b_j}^m}. \quad (7)$$

$$CP_k = \frac{1}{N_k} \sum_{l=1}^{N_k} g_{k,l}, \quad k = 1, 2, \dots, N_c. \quad (8)$$

Where $PI_{b_j}^m$ and $CP_{b_j}^m$ are the j^{th} histogram bin in the m^{th} sub-block of the probe image (PI) and class-prototype images, respectively. CP_k is the class-prototype of k^{th} class of subject. N_c is the number of classes and N_k is the number of gallery samples belonging to k^{th} class ($g_{k,l}$) [5].

VI. EXPERIMENTAL RESULTS

In order to evaluate the performance of proposed face recognition technique and effectiveness of image sub blocks on the recognition accuracy, we adopt four popular AR, ORL, LFW and FERET databases. We conduct four experiments and apply the identification algorithm on the 128×128 pixel images in the databases [1].

A. Face databases

- **AR Database:** AR face database includes 2600 images of 100 individuals (50 men and 50 women) [14]. Each subject has 26 images taken at two different sessions in two weeks (13 images per session). The images in the database are affected by illumination variation, facial expression and partial occlusion. We have employed non-occluded images in session 1 as gallery set and non-occluded images in session 2 with appearance changes in different time as probe set. Sample images of one subject in AR database are shown in Fig. 3a [1].
- **ORL Database:** Olivetti research lab (ORL) database consists of 40 individuals with 10 images per subject and appearance variation due to illumination changes, different time of acquiring image, facial expressions (open/close eyes and smiling/not smiling), up to 20 degree tilting and scales [15]. We randomly used 5 samples per individual in the gallery set and the remaining 5 images per subject in the probe set. Thus, we have 200 images per set. Figure 3b shows gallery and probe image samples of one individual in ORL database [1].
- **LFW Database:** Labeled faces in the wild (LFW) database includes 13,233 web-downloaded images of 5749 individuals [16]. LFW-a is the aligned version of LFW. Images are affected by pose and illumination changes, occlusion, blur, low resolution, race and aging effect in real-world scenarios. Some samples of LFW-a images are shown in Fig. 3c. In this paper, we randomly select 20 subjects with 12 images and less than 30 degree pose variation. Gallery set consists of the first half of 12 samples per individual and probe set includes the rest.
- **FERET Database:** The Face Recognition Technology (FERET) program database [17] is a huge database consists of 14,051 grayscale images in different subsets based on various illumination, facial expression and pose conditions. In this experiment, we use subset ba, bj and bk of 200 individuals and one images per subject in each subset. Subset ba includes regular frontal images. Subset bj consists of alternative frontal images corresponding to ba set. Subset bk also contains frontal images corresponding to ba but with different lighting conditions. 400 images in subsets bj and bk are used as the gallery set and subset ba is the probe set. Fig. 3d shows some samples of the probe images [5].

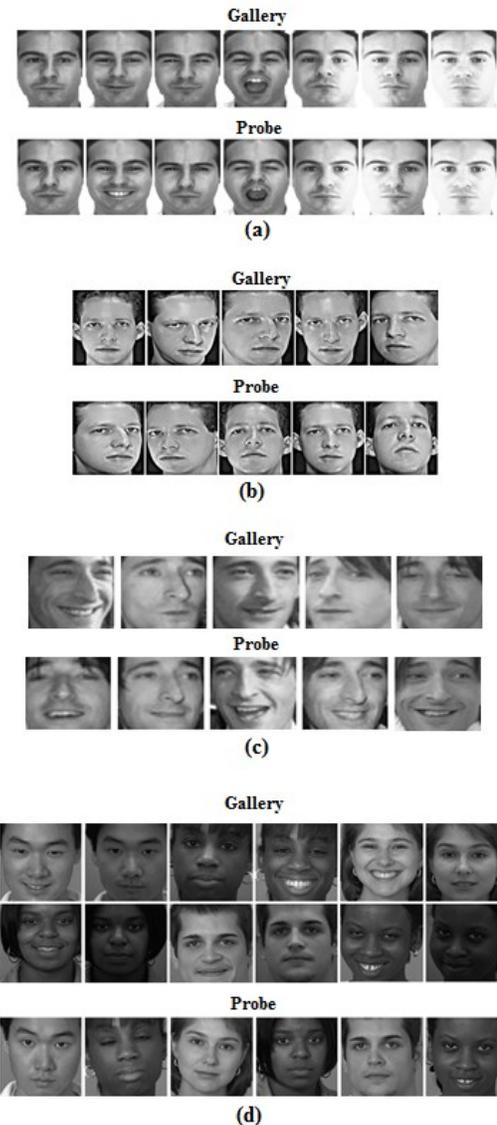


Figure 3. Sample images of one subject in (a) AR database, (b) ORL database, (c) LFW database, (d) FERET database.

B. Partial Recognition Based on the Image Subblocks

In this experiment we employ the proposed face recognition algorithm using an image sub-block at different locations and sizes. In order to find the effective size of selected sub-block, we find the accuracy of face recognition versus block size, which is shown in Fig. 4. It is shown that for all four databases, block size 32×16 pixels leads to the highest recognition accuracy. The location of the sub-block is near to the eye area. Fig. 5 shows the selected subregion for AR, ORL, LFW and FERET databases [1].

C. Decision Fusion for Selected Size of Subblock

Based on the results of previous section, the highest recognition accuracy is obtained at the block size of 32×16 pixels for four face databases. In this experiment, we employed the most effective block size and apply CSQ-weighted majority voting scheme by adding up the votes of

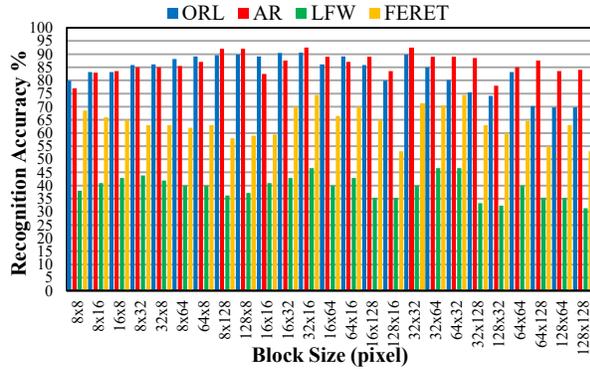


Figure 4. Recognition accuracy (%) of image subblocks for different block sizes.

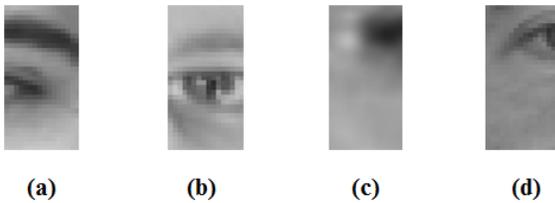


Figure 5. Location of the most effective image subregion: (a) AR database, (b) ORL database, (c) LFW database, (d) FERET database.



Figure 6. Sample images of four databases at different resolutions (128x128, 64x64, 32x32, 16x16 and 8x8 pixels, respectively from left to right): (a) AR database, (b) ORL database, (c) LFW database, (d) FERET database.

local classification results of image sub-blocks and finding the class label with maximum vote as the final decision. The result of sub-blocks fusion is shown in Table I and compared to the accuracy of other existing techniques, which shows the effectiveness of the proposed face recognition technique.

However, by employing the recognition process using only one sub-block of 32×16 pixels rather than the whole

image or fusion of local recognition results, the computational cost is reduced up to $\frac{1}{40}$ [1].

D. Effect of Low Resolution

In this experiment, based on the results of previous sections, we use the block size of 32×16 pixels for images of size 128×128 pixels and apply the proposed face recognition technique on image sub blocks and find the final decision using majority voting scheme. We aimed to evaluate the effect of low resolution images on the recognition accuracies of four databases. In order to verify the effect of resolution, we reduced the size of images from 128×128 to 64×64 , 32×32 , 16×16 and 8×8 pixels, respectively, and reduced the block sizes, relatively. Figure 6 shows images of four adopted databases at different resolutions. The recognition accuracy versus the image size is illustrated in Fig. 7. Reducing the resolution of images, degraded the accuracy of face recognition. However, the proposed technique shows relative stability against decreasing the resolution up to size of 32×32 pixels.

E. Effect of Partial Occlusion

In this experiment, we evaluated the effect of partial occlusions on the face, which included non-facial information, on the recognition accuracy. We conduct two experiments in this section as follows.

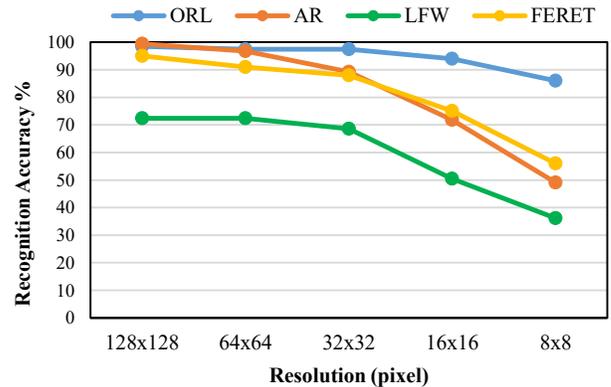


Figure 7. Recognition accuracy (%) of image subblocks for different block sizes.

TABLE I. RECOGNITION ACCURACY (%) OF DIFFERENT ALGORITHMS.

Block Size	Recognition Accuracy (%)			
	AR	ORL	LFW	FERET
LBP+MV [4]	93.42	95.50	63.81	95
CS-LBP+MV	80.42	91.50	53.33	56
LGBFR [5]	99	98	63.81	96.5
MS-CFB [6]	95	-	-	-
SADTF [7]	-	98.50	-	-
LCCR [18]	95.86	98	-	-
Proposed Method (Decision Fusion using weighted MV)	99.42	98.50	72.38	95

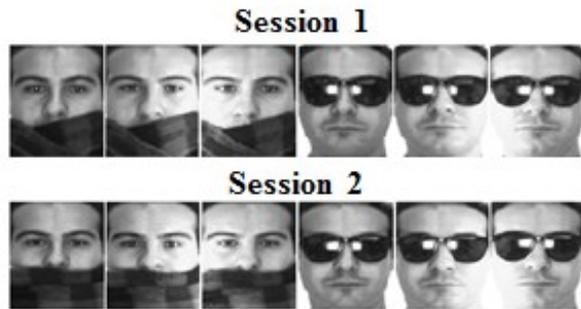


Figure 8. Occluded images of one subject in AR database at two sessions.

TABLE II. RECOGNITION ACCURACY (%) OF DIFFERENT ALGORITHMS.

Method	Sunglasses	Scarf
E-GV-LBP-M[19]	47.22	82.78
E-GV-LBP-P[19]	44.07	86.67
SADT[20]	95.5	75
Proposed Method (Decision Fusion using weighted MV)	88.83	97.17

- Real occlusion: AR database contains occluded images with sunglasses and scarf on the face, which are real occlusions. Figure 8 shows some occluded samples in AR database. We used gallery set of 200 images, by employing just two non-occluded images per subject with neutral expression in sessions 1 and 2. Two probe sets of 600 images consist of 6 occluded images per individual with scarf and 6 with sunglasses from both sessions, respectively. Table II shows the recognition accuracy of the proposed approach compared to the previous works.
- Unreal occlusion: In order to evaluate the performance of the proposed approach in the presence of incomplete information, we put a mandrill image and a black box at random positions on the images of the probe set of LFW database and varied the size of occlusion box from 0% to 90% of the complete image size (128x128 pixels). The accuracy percentage of identification versus the percentage of the occlusion box coverage is shown in Fig. 9. Figure 10 shows occluded images of one subject in the LFW database with different size of occlusion box. The recognition breakdown point of the mandrill occlusion occurs at 50% of the occlusion coverage while for the black box it occurs at 70% coverage, from where the identification accuracy decreases drastically. This is due to the fact that mandrill image contains facial components similar to the human's which leads to the misclassification.

VII. CONCLUSION

A block based face recognition algorithm has been proposed in this paper by dividing the magnitude and phase responses of Gabor filtered images. CS-LBP is applied on image sub-blocks and concatenation of local histograms at different scales and orientations gives the features of image sub-regions. Weighted majority voting is applied to combine the local decisions, made by sparse classifiers, which leads to the final decision on the identification of unknown individuals. Chi-square distance measurements are adopted as the local weights. The proposed approach outperforms the previous works for AR, ORL, LFW and FERET databases. Evaluating the recognition accuracy of different sub-regions of the face images gives the size and location of the most effective local area, which reduces computational complexity up to 2.5% and is very close the eyes area [1]. Moreover, the performance of proposed technique is verified versus the presence of incomplete information by resolution reduction and partial real and unreal occlusion. Reducing the resolution of images, degrades the accuracy of face recognition. However, the proposed technique shows relative stability against decreasing the resolution up to size of 32x32 pixels and the reduction in the identification accuracy is not noticeable. The proposed technique outperforms the previous works for real occlusion by scarf and sunglasses on images in the AR database. Based on the results partial recognition experiment, eyes area is the most effective sub region. Covering the eyes area by sunglasses leads to more reduction in the recognition accuracy. In addition, for artificial occlusion, occluding the image by more than 70% of the face area with black box and more than 50% with mandrill image, reduces the identification accuracy drastically.

Although the recognition results in the presence of incomplete information are impressive, further research to improve the applicability of the proposed technique to more challenging scenarios is required. Moreover, adopting other weighting techniques in the decision fusion to reduce the effect of undesirable regions is a possible direction to extend this work.

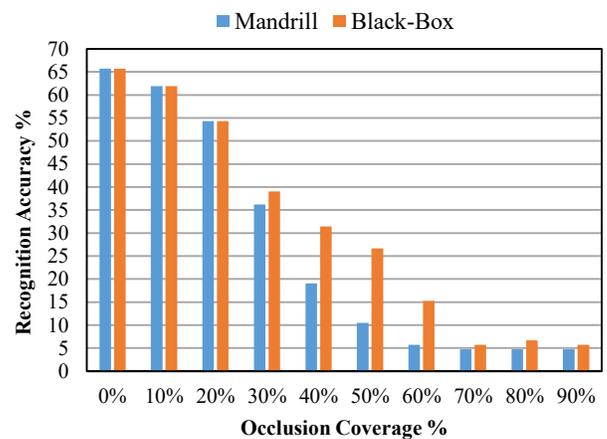


Figure 9. Recognition accuracy (%) of image subblocks for different block sizes.

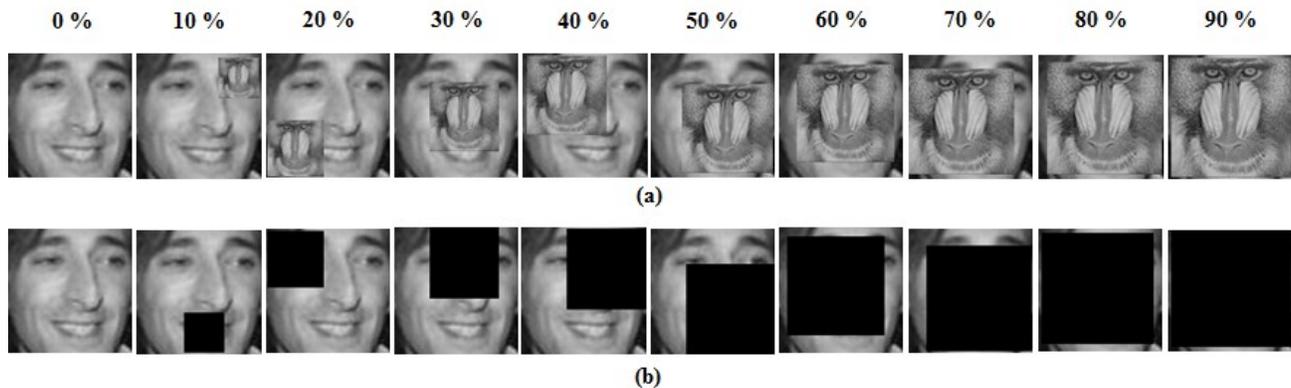


Figure 10. Recognition accuracy (%) of image subblocks for different block sizes.

REFERENCES

- [1] S. Nikan and M. Ahmadi, "Partial GMP-CS-LBP face recognition using image subblocks," Proc. International Conference on Systems (ICONS15), Barcelona, Spain, April 2015, pp. 36-39.
- [2] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, pp. 2454-2466, 2012.
- [3] W. Zhang, S. Shan, X. Chen, and W. Gao, "Local Gabor binary patterns based on Kullback-Leibler divergence for partially occluded face recognition," IEEE Signal Processing Letters, vol. 14, pp. 875-878, 2007.
- [4] S. Nikan and M. Ahmadi, "Human face recognition under occlusion using LBP and entropy weighted voting," Proc. International Conference on Pattern Recognition (ICPR12), Tsukuba, November 2012, pp. 1699-1702.
- [5] S. Nikan and M. Ahmadi, "Local Gradient-Based Illumination Invariant Face Recognition using LPQ and Multi-Resolution LBP Fusion," IET Image Processing, vol. 9, pp. 12-21, 2015.
- [6] Y. Yan, H. Wang, and D. Suter, "Multi-subregion based correlation filter bank for robust face recognition," Pattern Recognition, vol. 47, pp. 3487-3501, November 2014.
- [7] R. Mehta, J. Yuan, and K. Egiazarian, "Face recognition using scale-adaptive directional and textural features," Pattern Recognition, vol. 47, pp. 1846-1858, 2014.
- [8] S. Liao, A. K. Jain, and S. Z. Li, "Partial face recognition: alignment-free approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pp. 1193-1205, May 2013.
- [9] M. Heikkila, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," Pattern Recognition, vol. 42, pp. 425-436, 2009.
- [10] T. Ahonen, A. Hadid, and M. Pietikainen, "Face recognition with local binary patterns," Springer Computer Vision, vol. 3021, pp. 469-481, 2004.
- [11] Z. Lei, S. Liao, M. Pietikainen, and S. Li, "Face recognition by exploring information jointly in space, scale and orientation," IEEE Trans. Image Process., vol. 20, pp. 247-256, 2011.
- [12] S. Nikan and M. Ahmadi, "Classification fusion of global & local G-CS-LBP features for accurate face recognition," Proc. Int. Conf. on Testing and Measurement: Techniques and Applications (TMTA'15), Phuket, Thailand, pp. 303-307, January 2015.
- [13] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, pp. 210-227, 2009.
- [14] A. Martinez and R. Benavente, "The AR face database," CVC Technical Report, vol. 24, 1998. [Online]. Available from: <http://www2.ece.ohio-state.edu/~aliex/ARdatabase.html/> 2015.11.25.
- [15] The AT&T Laboratories Cambridge website, Available from: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html/> 2015.11.25.
- [16] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: a database for studying face recognition in unconstrained environments," Technical Report, University of Massachusetts, 2007.
- [17] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, pp. 1090-1104, 2000.
- [18] X. Peng, L. Zhang, Z. Yi, and K. K. Tan, "Learning locality-constrained collaborative representation for robust face recognition," Pattern Recognition, vol. 47, pp. 2794-2806, September 2014.
- [19] Z. Lei, S. Liao, M. Pietikainen, and S. Li, "Face recognition by exploring information jointly in space, scale and orientation," IEEE Transactions on Image Processing, vol. 20, pp. 247-256, 2011.
- [20] R. Mehta, J. Yuan, and K. Egiazarian, "Face recognition using scale-adaptive directional and textural features," Pattern Recognition, vol. 47, pp. 1846-1858, 2014.

Abstraction Layer Based Service Clusters Providing Low Network Update Costs for Virtualized Data Centers

Ali Kashif Bashir, Yuichi Ohsita, and Masayuki Murata
 Graduate School of Information Science and Technology
 Osaka University
 Osaka, Japan.
 e-mail: {ali-b, y-ohsita, murata@ist.osaka-u.ac.jp}

Abstract— Network virtualization is one of the most promising technology for the data centers. It was innovated to use the network resources efficiently by evaluating new protocols and services on the same hardware. This paper presents a virtual distributed architecture for virtualized data centers. This architecture group virtual machines according to their service types. Further, it deploys an abstraction layer for each group that helps in managing, controlling, and maintaining the group. An abstraction layer and a group of virtual machines together form a cluster. This abstraction layer offers several advantages, however, in this work, we considered network update cost when recovering from failures as a parameter of evaluation. Simulation results prove that the proposed architecture can detect and replace failed machines (virtual and physical) with low costs in comparison with the centralized approaches.

Keyword— network virtualization; service clusters; network failures; virtual data center network architecture; infrastructure for cloud applications; abstraction of virtual resources.

I. INTRODUCTION

This article is an extended version of our previous work [1]. It presents an architecture named Abstraction Layer based Service Clusters (AL-SC); in which virtual resources are grouped in clusters of various service types. Each cluster is managed and controlled by an Abstraction Layer (AL), which is a key element of this architecture and distinguishes it from the existing architectures. Deploying an AL offers several advantages to the virtual architecture; however, in this work we only evaluated the network update cost.

Network Virtualization (NV) [2] [3] [4] [5] [6] is one of the most promising technologies for the data centers (DCs). Introduced as a mean to evaluate new protocols and services [7]. It is already being actively used in research test-beds like G-Lab [8] or 4WARD [9], applied in distributed cloud computing environments [10]. Now, it is seen as a tool to overcome the obstacles of the current internet to fundamental changes. As such, NV can be thought of as an inherent component of the future inter architecture [11]. For DCs, it works as a backbone technology and let concurrent applications execute on a single hardware. Today, NV approaches are even applied in the telecommunication market, e.g., Open-Flow [12].

Virtualization is not a new concept. It is widely used to enhance the performance of DCs. With virtualization, we can create multiple logical Virtual Machines (VMs) on a single server to support multiple applications. These VMs takes the computation away from servers. VMware [13] and Xen [14] are two famous VMs proposed. However, virtualization of DC Networks (DCNs) aims at creating multiple Virtual Networks (VNs) at the top of a physical network [4]. Separation of VN/VNs from DCN offers several advantages, e.g., it allows to introduce customized network protocols and management policies. It lets concurrent applications run at the same underlying DCN and also help in securing the DC. On the other hand, without virtualization, we are limited to place a VM and also are limited in replacing or moving it.

VN, a primary entity in NV, is a combination of active and passive network elements (nodes and links) lies on top of a physical network. Virtual nodes are interconnected through virtual links, forming a virtual topology. With node and link virtualization, multiple VN topologies can be created and co-hosted on the same physical hardware. This virtualization introduces an abstraction that allows network operators to manage and modify networks in a highly flexible and dynamic way.

NV was envisioned to provide several features to the underlying infrastructure, e.g., scalability, flexibility, bandwidth improvement, etc. However, the existing virtual architectures provide only one or two features at a time. To enable the maximum features of NV for underlying architecture, in this paper, we propose AL-SC for DCs. The proposed architecture has two design aspects. First, it groups the VMs according to the service type they offer, e.g. VMs offering Map-reduce services can be grouped together. Note that, the number of services in an environment is defined by the network operator. Second, in order to manage, control, and monitor each groups, an AL is formed. It consists of a subset of VN switches that are separated with identifiers. A particular group of VMs and its' corresponding AL forms an SC. To the best of our knowledge, AL-SCs is the first proposed architecture that can be used in the service orchestration in the future.

An AL provides control to its cluster resources. Moreover, the number of switches in an AL are also

variable. Due to these, ALs offer several advantages to the architecture of AL-SC like scalability, flexibility, better management and control, etc. We have discussed some of these features in our previous works [1] [15]. In this work, we evaluated the effectiveness of AL-SC in terms of network update cost such as when recovering from network failures, e.g., VM or server failures. Evaluation results prove that AL-SC requires low cost in comparison with the centralized virtual approaches. Though, we believe that AL-SC performs better than the existing distributed approaches also, e.g., adaptive VN [16]; however, in this work, we did not provide the comparison.

The rest of the paper is organized as follows: in Section II, we presented the background of FI model and related works of the paper. In Section III, we discuss the overview, topology and a few other concepts of AL-VC. Section IV includes the mechanism to construct ALs. Section V includes our evaluation and Section 6 concludes the paper.

II. RELATED WORKS

DCs have gained a significant attention and rapid growth in both scale and complexity and are acting as a backbone for the cloud applications [17]. Companies like Amazon EC [18], Microsoft Azure [19], Facebook [20], and Yahoo [21] routinely use DCs for storage, search, and computations. In spite of their importance, architecture of today's DCs is far from being ideal due to following limitations:

- *No Performance Isolation:* Traditional DCs work as a single network and provide only best-effort solutions without performance isolation, which is required in modern cloud applications.
- *Inflexibility of the Network:* Due to non-flexible nature of traditional DCs, it is difficult to introduce new protocols or services. It leads to the minimal usage of the infrastructure.
- *Limited Management:* In the growing cloud application market, owners want control and need to manage the communication fabric for load-balancing, security, fault diagnosis, etc. However, the current architectures do not provide this flexibility.
- *Less Cost-effective:* Current DCs do not provide the support for multiple protocols. Applications usually require to migrate their management policies, VMs, and with the lack of support, the current infrastructure is not cost effective.
- *Internet Ossification* [22] [23]: The current Internet infrastructure is owned by a large number of providers, it is impossible to adopt a new architecture without the agreement of these stakeholders. Without consensus,

any initiative to improve Internet services will be difficult in nature and limited in scope.

NV is seen as a solution to all these problems. Virtualization of DCs resolves above mentioned issues, on the other hand, they are required to manage the physical infrastructure in the best possible ways. In literature, several solutions proposed for purpose. We will discuss the most relevant ones in this work. In [13], the authors surveyed on the importance of virtualization to improve flexibility, scalability, and resource utilization for data center networks. Whereas, MobileFlow [24] introduces carrier-grade virtualization in EPC. Diverter [25] is a software based network virtualization approach that does not configure switches or routers. It logically partition IP networks for better accommodations of applications and services. VL2 [26] is a data center network architecture that aims at achieving flexibility in resource allocation. In VL2, all servers belonging to a tenant share a single addressing space regardless of their physical location meaning that any server can be assigned to any tenant.

SecondNet [27] focused on providing bandwidth guarantees among VMs in a multi-tenant virtualized DC. It assumes a VDC (Virtual Data Cluster) manager that created VDCs. This work achieves high scalability by moving information about bandwidth reservation from switches to hypervisors. It also allows resources to be dynamically allocated and removed from VDCs. Another VN architecture, CloudNaas [28] provides support for deploying and managing enterprise applications in the clouds. It relies on OpenFlow forwarding [12]. CloudNaas provides several techniques to reduce the number of entries required in each switch. CloudNaaS also supports online mechanisms for handling failures and changes in the network policy specification by re-provisioning the VDCs. In NetLord [29], a tenant wanting to run a Map-Reduce task might simply need a set of VMs that can communicate via TCP. On the other hand, a tenant running a three-tier Web application might need three different IP subnets, to provide isolation between tiers. Or a tenant might want to move VMs or entire applications from its own datacenter to the cloud, without needing to change the network addresses of the VMs.

PolyVine [30] and adaptive VN [16] are two more worth discussing distributed approaches. Polyvine embeds end to end VNs in decentralized manners. Instead of technical, it resolves the legal issues among infrastructure providers. In adaptive VNs [16], every server is supposed to have an agent. Each server agent communicates with another to make local decisions. This approach is expensive and needs additional hardware.

All the above mentioned approaches are usually application specific and discuss one objective at a time. There is hardly any approach that enable set of NV features to the underlying infrastructure. AL-SC tends to fill this wide and provides several features, some of them are discussed below.

- *Scalability and Flexibility:* Due to the distributed nature, SCs are easy to manage, monitor, and update. Each cluster can be managed independently without interrupting the operation of the network.
- *Facilitate service chaining:* Network Service Chaining (NSC) [31] [32], an emerging direction in NV, can be easily implemented on our service clusters.
- *Increase network administration control:* In Software Defined Networking (SDN) environments, where users need to have the control of the network to write the applications. Having ALs can help the network manager to hide underlying infrastructure.
- *Efficient Query Allocation:* Another advantage of service clustering is that it can save the search and allocation time of Virtual Network Requests (VNRs). When virtual resources are exclusively grouped in clusters, VNRs can quickly find their desired VMs.

Note that, some of these features of AL-SC are discussed in our previous works and some we plan to discuss in the future.

III. SYSTEM OVERVIEW

This section discusses the overview of AL-SC. This work does not contain any VN mapping algorithm. However, the algorithm can be adopted from the literature, e.g., [33] [34] [35]. In [33] a VN mapping algorithm is provided that maps the VNs to underlying physical network in distributed and efficient manners. In [34] VN mapping algorithm also meet the bandwidth demands. There are many other algorithms exists in the literature. Any of that can serve the purpose. Therefore, in this work, we assume that VMs are already mapped at the hosts. Table I includes the list of the abbreviations used in this paper.

TABLE I. USED ABBREVIATIONS

Acronym	Descriptions
AL-SC	Abstraction Layer based Service Clusters
AL	Abstraction Layer
NV	Network Virtualization
DC	Data Center
VM	Virtual Machine
DCN	Data Center Network
VN	Virtual Network
VM	Virtual Machine
SDN	Software Defined Networking
VNR	Virtual Network Request
SC	Service Cluster
SNS	Social Networking Service
NM	Network Manager
OPS	Optical Packet Switch
NSC	Network Service Chain

A. Architectural Overview

Service Clusters (SCs) are more desirable than physical DCs because the resource allocation to VC can be rapidly adjusted as users' requirements change with time [27]. In DCs, two servers providing similar service have high data correlation in comparison with servers providing different service [28]. This property is also reflected in their VMs. In other words, in order to execute one VNR, two machines (servers/VMs) offering similar services are likely to interact with each other more. Therefore, one motivation behind grouping VMs into SCs is to save the VNR allocation time. Logical representation of AL-SC is shown in Figure 1, where a DCN is virtualized into VCs of different service types, i.e., VC of Social Networking Services (SNSs), VC of Web services, VC of map-reduce, etc. This architecture can be implemented in several other ways. For example, in the environment where a single or multiple virtualized DCN are owned by multiple network operators. In that case, each operator can manage, control, and modify its own virtual resources in the shape of SCs. Classification of clusters according to the service or traffic type can be used in service orchestration in the future.

B. Topology

Ideally, VN topology should be constructed in a way that it achieves minimum energy consumption and larger bandwidth without delay. Minimum energy consumption can be achieved by minimizing the active number of ports and constructing energy efficient routes. Larger bandwidth can be achieved by adding virtual links in the VN and by managing traffic efficiently. Delay can be improved by using efficient routes and by processing data faster at switches. We argue that the proposed architecture has potential to provide all these features.

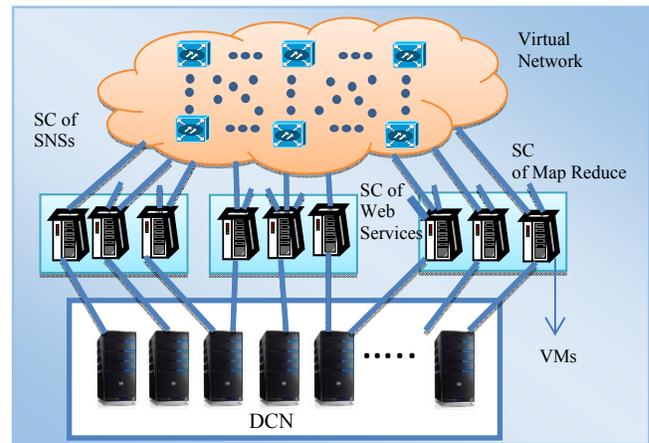


Figure 1. Overview of AL-SC.

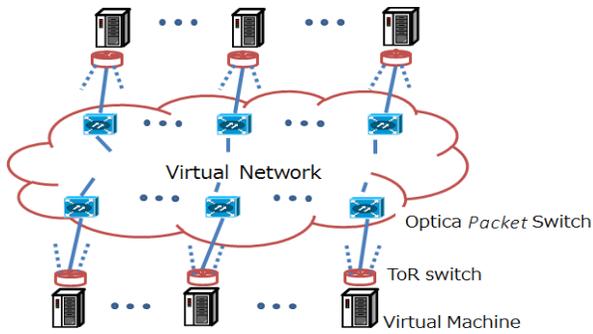


Figure 2. Construction of VN.

Connectivity of AL-SC is presented in Figure 2, where all the servers in a server rack are connected to one Top-of-the-Rack (TOR) switch. Each server is hosting multiple VMs. In the core of the network, to construct virtual links, we use Optical Packet Switches (OPSs). TOR switches produce electronic packets and they need to be converted into optical packets before sending over the optical domain of the network. Optical packets will be converted back to the electronic packets before forwarding to the TOR switches. This electronic/optical/electronic conversion is costly and should be reduced to increase the network performance. To read further on this, readers are suggested to read [36].

In AL-SC, every VM is connected to multiple OPSs. OPS that joins a particular AL can have four possible types of connections, namely: 1. with TOR switches, 2. with VMs of local cluster, 3. with OPSs of local AL, and 4. with OPSs of VN that are not part of its local AL. In Figure 3, we also presented the block diagram of this connectivity and as well the logical construction of AL-SC. After defining the

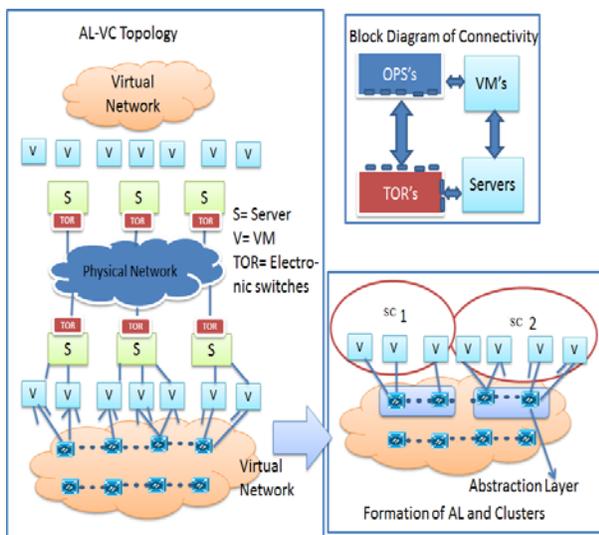


Figure 3. AL-SC Topology.

topology, we would like to discuss the communication pattern in this architecture. Categorically, this communication is divide into two as:

Intra-cluster Communication: Communication is intra-cluster when the destination VMs service type is the same as sender VM, which means it exists in the same SC. Most of the intra-cluster communication consists of small, but a large number of queries. AL- SC provides shorter routes to this traffic to reduce latency. Local switches of the corresponding AL of the particular cluster can interact with each other to find the destination VM, as shown in Figure 4.

Inter-cluster communication: Communication is inter-cluster when the destination VM belongs to another SC. This communication is usually less common than an intra-cluster. On the other hand, it generates a huge amount of traffic, hence, require high bandwidth, e.g., VM migration. Providing higher bandwidth is one of the characteristics of optical domain. Therefore, we construct optical paths in the VN as shown in Figure 4. We also have allocated a set of switches in an AL to handle inter-cluster traffic. Handling most of the queries (intra-cluster) within a cluster will motivate to providing dedicated paths for inter-cluster traffic.

C. Optical Packet Switches

Proposed topology can be constructed using packet switches. However, in order to achieve higher bandwidth with small energy consumption, we use OPS [37]. We modified the structure of OPS proposed by Urata et al. [38] as shown in Figure 5, where OPS is constructed of multiple wavelengths. Optical packets from other OPSs are demultiplexed into optical signals of each wavelength. After label processing, these packets are relayed to the destination port. A shared buffer is constructed of CMOS. It stores the packets in case of collision or when packets received from

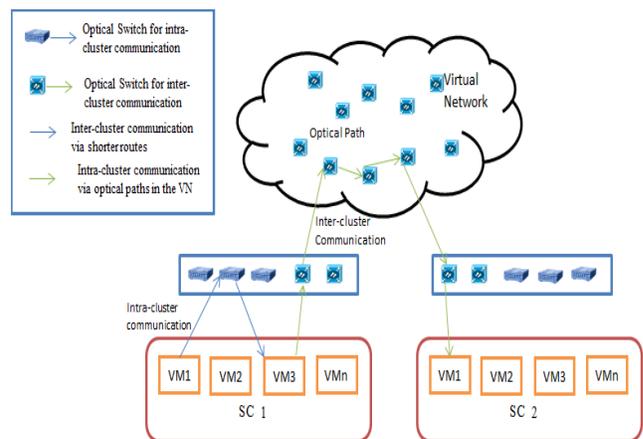


Figure 4. Communication in AL-SC.

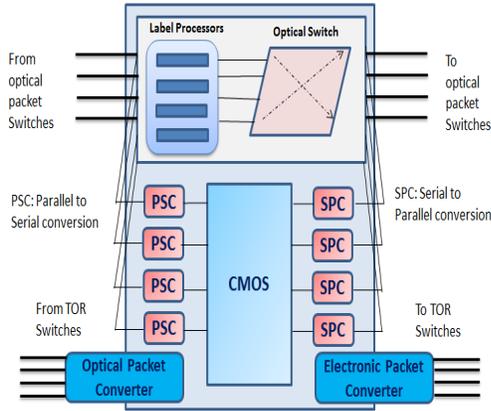


Figure 5. Structures of Optical Packet switch.

TOR switches. Packets from TOR switches are aggregated, stored in the buffer, and relayed to OPSs after parallel-to-serial conversion. Similarly, optical packets that need to be forwarded to TOR switches are converted to electronic packets. In simple words, this switch converts electronic packets to optical and vice versa. Detailed topology of AL-SC is explained in the next section.

D. Network Manager (NM)

A central NM is one of the most important entities, which is responsible for the operation of AL-SC. It decides the number of SCs, size of SCs, SC formation logic, and how they are mapped to the underlying DCN. Moreover, it also manages the physical resources like servers, VMs, links, etc. NM is responsible for VC formation and deletion. It also assigns each SC with a unique SCID and IP address. However, controlling and managing the cluster after creation is the job of its AL. In the future, ALs can be controlled by different application with the help of SDN. For address isolation, every SC has its own IP address space.

IV. ABSTRACTION LAYER

In the previous section, we presented the overview of AL-SC and highlighted its design aspects. In this section, we would like to present the AL construction algorithm.

A. Construction of an AL

The basic idea behind the construction of an AL is logically assigning a subset of the OPSs to a particular group of VMs. Group of VMs and an AL together is called a cluster in this work. In an AL, we assume every OPS knows the topology of its cluster, such as locations of VMs and their connections.

To construct an AL, VMs of every cluster selects the minimum subset of OPSs that connects them. This approach first selects the OPSs with highest connections and then OPSs with second highest connections and so on until all the VMs are connected. Finally, the subset of OPSs that covers all the VMs of a cluster will be declared as its AL. Switches of an AL will be differentiated from other OPSs of VN with the respective cluster ID. Information of these switches such as switch ID and IP addresses is forwarded to all the VMs. This procedure is repeated for every cluster until all the clusters have an AL.

Selecting switches with maximum connections reduce the number of switches in an AL, which will help in filtering and aggregating the traffic. On the other hand, it will increase the overhead at certain switches and will result in congestion. For that, we need to make sure that an AL has significant number of switches to handle congestion and to meet the required bandwidth demand. To meet these challenges, more refined algorithms are planned to be proposed in the future. The detailed mechanism of the current algorithm is as follows:

Step 1: As mentioned earlier, we assume that VMs are already mapped to servers and are grouped into clusters according to their service types. After this grouping, they connect themselves to the switches of VN. These connections can be established randomly or based on a specific criterion. In this work, we use random approach shown in Figure 6. The selection probability of the OPSs of AL is based on the distance, in which, we have

$$P_i = \frac{R_i}{\sum_j d_j} \tag{1}$$

Where

P_i = probability of selecting switches v_s^i
 d_j = distance of switches from VM

Step 2: Each VM sends a list of the OPSs they connect to the NM. Figure 7(a) shows this list.

Step 3: NM selects the minimum set of OPSs that cover all VMs of a group. To explain this, let's assume a graph $G = (V, E)$ with links $l_i \geq 0$, where the objective is to find a

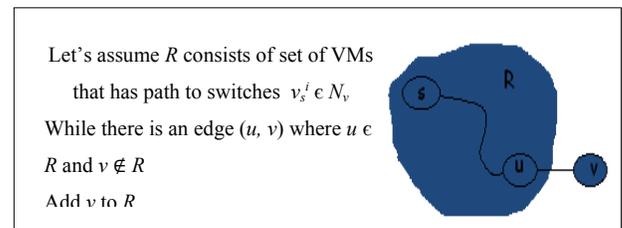


Figure 6. Switch selection criteria.

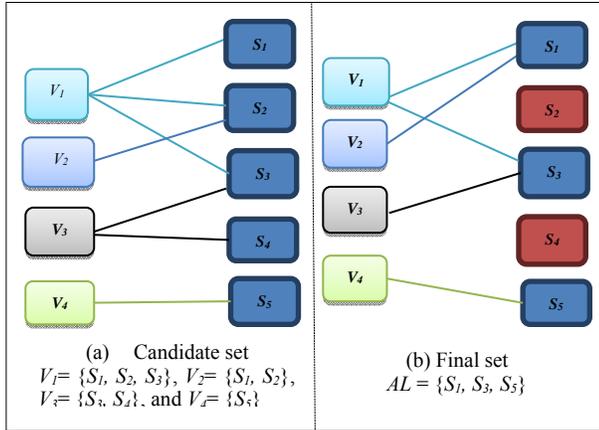


Figure 7. Selection of an AL.

minimum subset of switches that covers all VMs. For this, we apply the following condition to VMs

$$S_i = \begin{cases} 0 & \text{if VM } v_s^i \text{ is not covered} \\ 1 & \text{if VM } v_s^i \text{ is covered} \end{cases}$$

Objective function: minimize $\sum I/S$ for all v_s

Figure 7(b) is the final minimum subset of the OPSs required to form an AL for a cluster. These switches such as S_1 , S_2 , and S_3 in Figure 7(b), will be announced as an AL for a cluster. S represent an OPS. These OPSs will be assigned with SC_{ID} . In routing the traffic, OPSs in the intra-cluster phase can be addressed with (S_{ID} , and IP address) and in inter-cluster phase as (S_{ID} , SC_{ID} , and IP address).

Step 4: After selecting an AL, the remaining candidate switches will be discarded and they continue being part of the core of the VN.

This procedure is repeated until every cluster has an AL.

V. EVALUATION

NV plays a crucial role for DCNs. Since last few years, it is one of the most widely researched topics in cloud computing. Several architectures and solutions proposed that virtualize the physical resources. In all these works, underlying physical topology of the DCN plays an important role in the performance of virtual architectures as they provide the real grounds. Most of the existing virtual architectures are implementable on one or few topologies; however, AL-SC can be implementable on several topologies, such as B Cube, VL-2, FATTree, etc. It collects virtual resources from the underlying topology and group them according to the administrative logic as shown in the Figure 8. However, for evaluation of AL-SC, we choose

TABLE II. ENVIRONMENT

Number of Servers	96
Number of VMs	360
Max VM a server can host	10
Number of switches in AL	10 % of VM in the cluster
Number of clusters	2, 4, 6, 8, and 10
DCN topology	FATREE
Parameters	Average time and Communication Cost

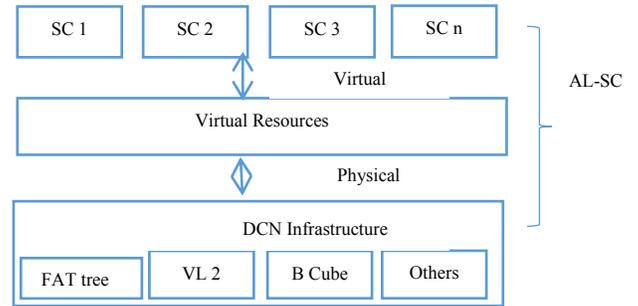


Figure 8. Implementability of AL-SC.

FATTree [39] as underlying topology. In this work, we will evaluate the network update cost of AL-VC in terms of deletion or failure of VMs and in terms of addition of VMs. The cost is measured in the number of messages and time required. Therefore, we measured the communication cost in terms of messages and time when VMs are added or deleted in AL-SC architecture. Our evaluation environment is presented in Table II.

A. Network Update Cost in finding new VM

In this evaluation, we measured the time and communication cost in order to find a new VM. Several scenarios can be considered in this aspect. For example, migration of a VM from one server to another or from one cluster to another, failure or deletion of VM. In every scenario, our algorithms consist of following steps:

VM Discovery Mechanism

- i. A VM is considered failed or migrated when it is not replying to the control messages of its AL.
- ii. AL informs all the VMs of its cluster about this failure or movement with the ID of VM.
- iii. AL will request the server whose VM is failed to host a new VM.
- iv. If the server does not have enough resources to host a new VM, it will send attributes of the failed VM to the AL.
- v. AL will request other servers that have the resources to host new VM.
- vi. Servers will send the attributes of candidate VMs to AL.

- vii. AL will select the server that has VM with the attributes most closed to fail VM.
- viii. Finally, the failed VM will be replaced with a new VM.

The attributes of the requested VM can be represented as:

$$\alpha\tau\tau_{N\sigma} = ((\alpha\tau\tau_1, v_{\sigma}^1), (\alpha\tau\tau_2, v_{\sigma}^2), \dots, (\alpha\tau\tau_v, v_{\sigma}^v)) \quad (2)$$

Non-functional (NF) attributes of the two VMs can be calculated by the following dissimilarity metrics.

$$dism(i, j) = \frac{\sum_{r=1}^l \delta_{ij}^r}{\sum_{r=1}^l \delta_{ij}^r} \quad (3)$$

Where

l is the number of NF attributes

δ_{ij}^r denotes the dissimilarity of VM i and j related to $\alpha\tau\tau_r$.

δ_{ij}^r expresses the coefficients of the NF attributes of machines i and j .

In Figures 9, 10, and 11, we evaluated the performance of AL-SC in detecting and replacing failed or migrated VM/VMs. To measure the performance of our algorithm, we constructed various cluster sizes in the network, e.g., 2, 4, 6, 8, and 10. Since we have no distributed control in the centralized approach, therefore, the detection of VMs will happen at the NM. For that, the central entity exchanges messages with all participating servers to discover a new server to host the new VM. However, in AL-SC this mechanism is executed within the cluster, i.e., on its AL. AL requires the fewer of messages and less time to find a new VM in comparison with centralized scheme.

In Figure 9, we measured the time required to replace a VM. In this evaluation, we consider three cases: the best, average, and the worst. In all three cases, we only measured the response time of machines in order to replace the VM. Whereas in Figure 10 represents the communication cost in terms of the number of messages required for this replacement. From these figures, we can see that an increased number of clusters decrease the average time and communication cost. This is because the number of participating entities in finding a new VM decrease. The increasing number of clusters helps in improving the performance of our algorithm. On the other hand, too many clusters may result in increased overhead, hence, a trade-off exists.

In Figure 11, we measure the time to replace the multiple failed VMs. The detection mechanism will remain same as mentioned above. It is clearly seen that performance of AL-SC significantly better than the centralized approach. Without SCs, NM has a lot of workload as a result of multiple failure detection and replacements. In case of AL-SC, each AL can run this mechanism locally the VM discovery procedure to find the new VMs with less overhead.

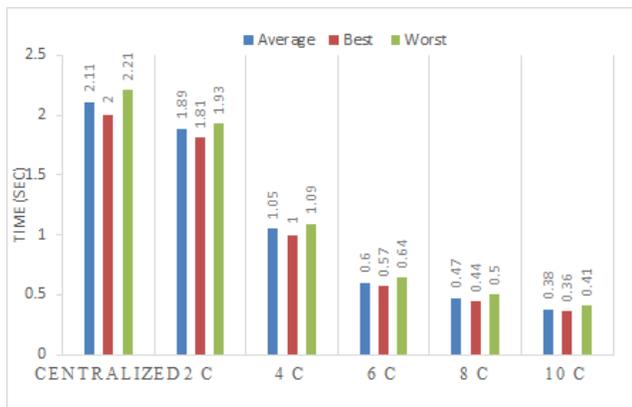


Figure 9. Time required to replace the failed VM.

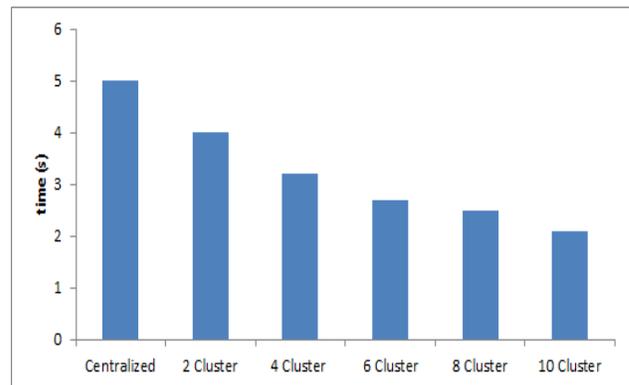


Figure 10. Communication Cost require in replacing the failed VM.

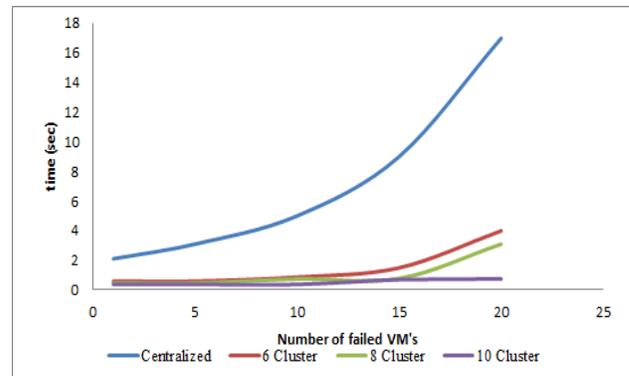


Figure 11. Average time required to replace the failed VMs.

B. Addition of VM

The architecture of AL-SC supports the addition of new VMs. In Figure 12, we evaluated the time required to add a new VM in the best, the average, and the ideal modes. The evaluation is conducted assuming no delay in the network. From the results, it is clear that if we have more refined clustering, i.e., more service types, it will help in saving the time required to add a VM. The algorithm will be as follows.

- I. A VM request NM with a join message.
- II. NM collects the NF attributes of the VM and matches it with the server attributes.
- III. Server with the closest NF attributes will be requested to host the VM. If the server resources are limited, then the second closest NF server will be requested. This will continue until the host server is found.
- IV. After joining a server, the VM will request to join the AL on the base of service type.
- V. AL accepts the new VM and update its cluster topology and send the joining message with the ID of new VM to all existing VMs.

C. Network Update cost in terms of Server Failure

When a server fails, all the VMs hosted by that server will be considered failed. Therefore, to keep the operation of the SCs, new hosts for these VMs need to be found. The procedure of discovery will be as follows

- I. If a VM does not respond to keep-alive messages, AL considers it failed and contact with its server.
- II. If the server also does not respond, AL assumes that the server has failed.
- III. As servers are physical resources of a DCN, an AL does not keep the attributes of the server.

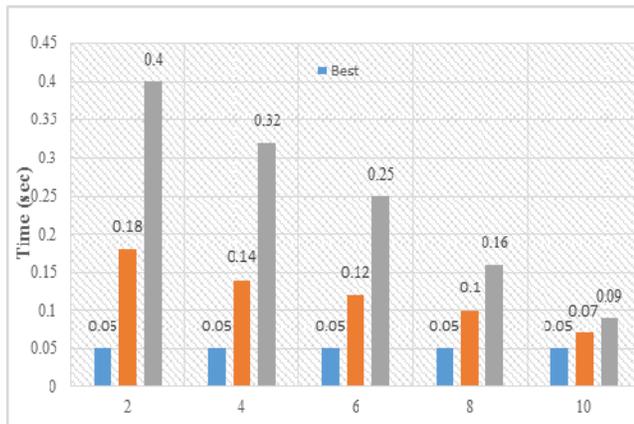


Figure 12. Time required to add a VM.

- IV. Therefore, it informs the NM and asks for the attributes of the failed server and its VMs. VMs attributes are stored in both the NM and in the hosting servers. However, server attributes can be fetched only from the NM.
- V. After receiving NF attributes, AL runs a local VM discovery algorithm to find new hosts for the VMs as explained before.

In AL-SC, if the failed servers and VMs belong to only one cluster, it will not affect the operation of other clusters. To evaluate the update cost, we assume that the failing server was hosting three VMs that require new host now. The update cost of finding new host/hosts for these VMs is calculated in Figure 13 and 14, in which we can clearly see that the cost decreases as the number of cluster increases. In this evaluation, we assumed that all three VMs of the failing servers were belonged to one SC. Let us consider the case when VMs belongs to more than only one SC. In this situation, average time to find new host will remain same; however, the number of messages required, will increase as the VM attribute matching algorithm will run in the multiple SCs.

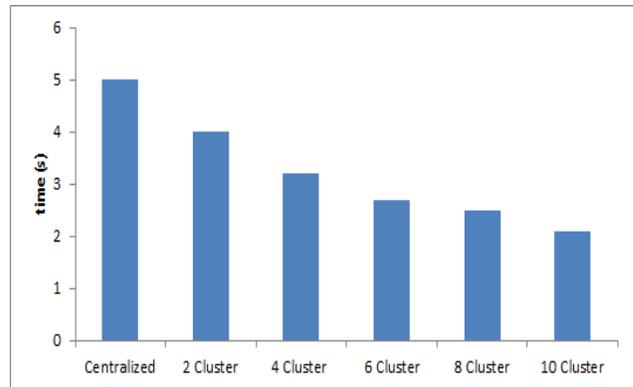


Figure 13. Average time to recover from a Server failure.

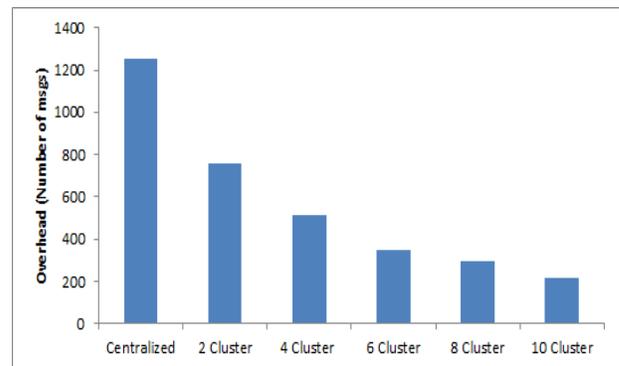


Figure 14. Communication cost require recovering from a server failure.

In case, the servers of a particular SC have no more resources to host new VMs, new servers will be requested from the NM. NM request the infrastructure provider to deploy new servers. They will be deployed in the infrastructure by joining the non-Functional attributes of the failing server. Attributes of the requested server can be represented as follows.

$$\alpha\tau\tau_{N\sigma} = ((\alpha\tau\tau_1, v_{\sigma}^1), (\alpha\tau\tau_2, v_{\sigma}^2), \dots, (\alpha\tau\tau_v, v_{\sigma}^v)) \quad (4)$$

VI. CONCLUSIONS AND FUTURE WORKS

The existing infrastructure of the data centers has several limitations. Network virtualization helps in overcoming these limitations and enables the cloud applications for users. In this paper, a distributed virtual architecture is proposed that virtualizes data center into multiple service clusters according to the service types exist in the network. Each service cluster consists of a group of VMs and a group of virtual switches called abstraction layer. An abstraction layer is the subset of virtual network switches and offers several features to the virtualized architecture. In this work, we only evaluated the network update cost when the virtual or the server machine fails. From the results, we can see that this architecture requires less time and cost in detecting the failures and recovering from them.

We plan to extend this work in the multiple directions. First, we will propose a more efficient mechanism for the construction of abstract layer. Second, we will improve other parameters, e.g., bandwidth with AL-SC. We also plan to adopt this architecture in the network service chaining.

ACKNOWLEDGEMENT

This work was supported by the National Institute of Information and Communications Technology (NICT), Japan.

REFERENCES

- [1] A.K. Bashir, Y. Ohsita, and M. Murata, "Abstraction Layer Based Distributed Architecture for Virtualized Data Center," in Proceedings of the 6th International Conference on Cloud Computing, Grids, and Virtualization (CLOUD COMPUTING 2015) IARIA, Mar. 2015, pp. 46-51, ISBN: 978-1-61208-388-9.
- [2] N. M. M. K. Chowdhury, "Network virtualization: State of the art and Research Challenges," *Comm. Mag., IEEE*, vol. 47, no. 7, pp. 20-26, Jul. 2009, doi: 10.1109/MCOM.2009.5183468.
- [3] A. Berl, A. Fischer, and H. D. Meer, "Virtualisierung im Future Internet- Virtualisierungsmethoden Und anwendungen," *Info. Spek. Springer*, vol. 33, no. 2, pp. 186-194, Apr. 2010, doi: 10.1007/s00287-010-0420-z
- [4] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network Virtualization: Implementation steps towards the Future Internet," *Elec. Comm. EASST*, vol. 17, Mar. 2009, doi: <http://dx.doi.org/10.14279/tuj.eceasst.17.216>
- [5] D. Stezenbach, M. Hartman, and K. Tutschku, "Parameters and Challenges for Virtual Network Embedding in the Future Internet," in Proceedings of Network Operations and Management Symposium (NOMS 2012) IEEE, pp. 1272-1278, Apr. 2012, ISSN: 1542-1201, ISBN: 978-1-4673-0267-8
- [6] A. Berl, A. Fischer, and H. D. Meer, "Using System Virtualization to Create Virtualized Networks," *Elec. Comm. EASST*, vol. 17, Mar. 2009, ISSN: 1863-2122.
- [7] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet Impasse through Virtualization," *Comp. IEEE*, vol. 38, no. 4, pp. 34-41, Apr. 2005, doi: <http://doi.ieeeecomputersociety.org/10.1109/MC.2005.136>
- [8] D. Schwerdel, D. Günther, R. Henjes, B. Reuther, and P. Müller, "German-lab Experimental Facility," *Third Future Internet Symposium (FIS 2010)*, vol. 6369, Sep. 2010, ISBN: 978-3-642-15876-6, doi: 10.1007/978-3-642-15877-3_1
- [9] J. Carapina and J. Jiménez, "Network Virtualization: a view from the bottom," in Proceedings of the Workshop on Virtualized Infra Syst. and Arch (VISA 09) ACM, pp. 73-80, 2009, ISBN: 978-1-60558-595-6, doi:10.1145/1592648.1592660
- [10] P. Endo, A. Palhares, N. Pereira, G. Goncalves, D. Sadok, and J. Kelner, "Resource Allocation for Distributed Cloud: Concepts and Research Challenges," *Netw. IEEE*, vol. 25, no. 4, pp. 42-46, Jul. 2011, doi: 10.1109/MNET.2011.5958007
- [11] N. Feamster, L. Gao, and J. Rexford, "How to lease the internet in your spare time," *Comp. Comm. Rev. ACM SIGCOMM*, vol. 37, pp. 61-64, 2007, doi:10.1145/1198255.1198265
- [12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, and J. Rexford, "Openflow: Enabling Innovation in Campus Networks," *Comp. Comm. Rev. ACM SIGCOMM*, vol. 38, no. 2, pp. 69-74, Apr. 2008, doi:10.1145/1355734.1355746
- [13] M. F. Bari, R. Boutaba, R. Esteves, L.Z. Granville, M. Podlesny, and M.G. Rabbani, "Data Center Network Virtualization: A Survey," *Comm. Surv.& Tuto. IEEE*, vol. 15, pp. 909-928, May. 2013, doi: 10.1109/SURV.2012.090512.00043
- [14] VMware. "Virtualization Overview," White paper, 2006. Available from: <https://www.vmware.com/pdf/virtualization.pdf> 2015.12.03
- [15] A.K. Bashir, Y. Ohsita, and M. Murata, "A Distributed Virtual Data Center Network Architecture for the Future Internet," *Technical Reports of IEICE (IN2014-165)*, pp. 261-266, Mar. 2015.
- [16] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive Virtual Network Provisioning," in Proceedings of the 2nd Workshop on Virtualized Infrastructure Systems and Architectures (VISA 10) ACM SIGCOMM, pp. 41-48, Sep. 2010, ISBN: 978-1-4503-0199-2, doi: 10.1145/1851399.1851407
- [17] P. Endo, A.D. A. Palhares, N. Pereira, and J. Mangs, "Resource Allocation for Distributed Cloud: Concepts and Research Challenges," *Netw. IEEE*, vol. 25, no. 4, pp. 42-46, Jul. 2011, doi: 10.1109/MNET.2011.5958007
- [18] "Amazon Web Services: Overview of Security Processes," Amazon, pp. 1-75, Aug. 2015, Available from: <https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf>, 2015.12.03
- [19] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, and S. McKelvie, "Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency," *23rd Symposium on Operating System Principles (SOSP 11) ACM*, pp. 143-157, 2011, ISBN:978-1-4503-0977-6 doi: 10.1145/2043556.2043571

- [20] N. Farrington and A. Andreyev, "Facebook's Data Center Network Architecture." Facebook, Inc, Available from: <http://nathanfarrington.com/papers/facebook-oic13.pdf>, 2015.12.03
- [21] Y. Chen, S. Jain, V. K. Adhikari, Z.L. Zhang, and K. Xu. "A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets," in Proceedings of the INFOCOM 2011, pp. 1620-1628, Apr. 2011, ISBN: 978-1-4244-9919-9, doi: 10.1109/INFCOM.2011.5934955
- [22] J. S. Turner and D. E. Taylor, "Diversifying the Internet," in Proceedings of the Global Telecommunication Conference (GLOBECOM 05) IEEE, Dec. 2005, ISBN: 0-7803-9414-3, doi: 10.1109/GLOCOM.2005.1577741
- [23] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A Survey of Green Networking Research," *Comm. Surv. & Tut. IEEE*, vol. 14, pp. 3-20, Feb. 2012, doi: 10.1109/SURV.2011.113010.00106
- [24] K. Pentikousis, Y. Wang, and W. Hu, "MobileFlow: Toward Software-Defined Mobile Networks," *Communications Magazine*, IEEE, vol. 51, no. 7, pp. 44-53, July 2013
- [25] A. Edwards, F. A. and A. Lain, "Diverter: A New Approach to Networking Within Virtualized Infrastructures," in Proceedings of the 1st Workshop on Research on Enterprise Networking (WREN 09) ACM, pp. 103-110, Aug. 2009, ISBN: 978-1-60558-443-0, doi: 10.1145/1592681.1592698
- [26] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, and P. Lahiri, "VL2: A Scalable and Flexible Data Center Network," in Proceedings of the Data Communication Conference (SIGCOMM) ACM, pp. 51-62, Oct. 2009, ISBN: 978-1-60558-594-9 doi: 10.1145/1592568.1592576
- [27] C. Guo, G. Lu, J. H. Wang, S. Yang, C. Kong, and Y. Zhang. "SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees," in Proceedings of the 6th International Conference (Co-NEXT 10), ACM, 2010, ISBN:978-1-4503-0448-1 doi: 10.1145/1921168.1921188
- [28] T. Benson, A.A.A. Shaikh, and S. Sahu, "CloudNaaS: A Cloud Networking Platform for Enterprise Applications," in Proceedings of the Symposium on Cloud Computing (SOCC 11) ACM, Oct. 2011, ISBN: 978-1-4503-0976-9, doi: 10.1145/2038916.2038924
- [29] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters," in Proceedings of the SIGCOMM, ACM, pp. 62-73, Aug. 2011, ISBN: 978-1-4503-0797-0, doi:10.1145/2018436.2018444
- [30] M. Chowdhury, F. Samuel, and R. Boutaba, "PolyViNE: Policy-based Virtual Network Embedding across Multiple Domains," in Proceedings of the Workshop on Virtualized Infrastructure Systems and Architectures (VISA 10) ACM SIGCOMM, pp. 49-56, Sep. 2010, ISBN: 978-1-4503-0199-2, doi: 10.1145/1851399.1851408
- [31] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Optical Service Chaining for Network Function Virtualization Communication," *Maga. IEEE*, vol. 53, issue 4, pp. 152-158, Apr. 2015, doi: 10.1109/MCOM.2015.7081089J.
- [32] M. Sanner, M. Ouzzif, and Y.H. Aoul, "DICES: a Dynamic Adaptive Service-driven SDN Architecture," in Proceedings of the Network Softwarization Conference (NetSoft) IEEE, pp. 1-5, Apr. 2015, doi: 10.1109/NETSOFT.2015.7116125
- [33] J.F. Botero, X. Hesselbach, A. Fischer, and H.d. Meer, "Optimal Mapping of Virtual Networks with Hidden Hops," *Tele. Syst. Springer*, vol 51, no. 4, pp. 273-282, Dec. 2012, doi: 10.1007/s11235-011-9437-0
- [34] I. Houidi, W. Louati, and D. Zeghlachie, "A Distributed Virtual Network Mapping Algorithm," in Proceedings of the International Conference on Communications (ICC 08) IEEE, pp. 5634-5640, May 2008, ISBN: 978-1-4244-2075-9, doi: 10.1109/ICC.2008.1056
- [35] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey" *Comm. Surv. & Tuto. IEEE*, vol. 15, no. 4, pp. 1888-1906, Nov. 2013, doi: 10.1109/SURV.2013.013013.00155
- [36] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network Function Placement for NFV Chaining in Packet/Optical Data Centers," *Light Wave Tech.*, vol. 33, no. 8, Apr. 2015.
- [37] Y. Ohsita and M. Murata, "Data Center Network Topologies using Optical Packet Switches," in Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW) IEEE, pp. 57-64, Jun. 2012, ISBN: 978-1-4673-1423-7, doi: 10.1109/ICDCSW.2012.53
- [38] R. Urata, T. Nakahara, H. Takenouchi, T. Segawa, H. Ishikawa, and R. Takahashi, "4x4 Optical Packet Switching of Asynchronous burst Optical Packets with a Prototype, 4x4 Label Processing and Switching Sub-system," *Opt. Expr.*, vol. 18, no. 15, pp. 15283-15288, Jul. 2010, doi: 10.1364/OE.18.015283.
- [39] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," in Proceedings of the Conference on Data Communication (SIGCOMM 08) ACM, pp. 63-74, 2008, ISBN: 978-1-60558-175-0, doi: 10.1145/1402958.1402967
- [40] Y. Zhang, A.J. Su, and G. Jiang, "Evaluating the Impact of Data Center Network Architectures on Application Performance in Virtualized Environments," in Proceedings of the 18th International Workshop on Quality of Service (IWQoS) IEEE, pp. 1-5, Jun. 2010, ISBN: 978-1-4244-5987-2, doi: 10.1109/IWQoS.2010.5542728

Looking for the Best, but not too Many of Them: Multi-Level and Top-k Skylines

Timotheus Preisinger

DEVnet Holding GmbH
Grünwald, Germany

Email: t.preisinger@devnet.de

Markus Endres

University of Augsburg
Augsburg, Germany

Email: markus.endres@acm.org

Abstract—The problem of *Skyline* computation has attracted considerable research attention in the last decade. A *Skyline* query selects those tuples from a dataset that are optimal with respect to a set of designated preference attributes. However, the number of *Skyline* answers may be smaller than required by the user, who needs at least k . Given a dataset, a *top-k Skyline* query returns the k most interesting elements of the *Skyline* query based on some kind of user-defined preference. That said, in some cases, not only the Pareto frontier is of interest, but also the stratum behind the *Skyline* to get exactly the *top-k* objects from a partially ordered set stratified into subsets of non-dominated tuples. In this paper we present the concept of *multi-level Skylines* for the computation of different strata and we discuss *top-k Skyline* queries in detail. Our algorithms rely on the lattice structure constructed by a *Skyline* query over low-cardinality domains. In addition we present external versions of our algorithms such that it is not necessary to store the complete lattice in main memory. We demonstrate through extensive experimentation on synthetic and real datasets that our algorithms can result in a significant performance advantage over existing techniques.

Keywords—*Skyline*; *Preferences*; *Multi-level*; *Top-k*; *Lattice*.

I. INTRODUCTION

Information systems of different types use various techniques to rank query answers. In such systems users are often interested in the most important (top-k) and most preferred query answers in the potentially huge answer space. Different preference-based query languages have been defined to support the bases for discriminating poor quality data and to express user's preference criteria on top-k [1][2][3].

The *Skyline* operator for example [4] has emerged as an important and popular technique for searching the best objects in multi-dimensional datasets. A *Skyline* query selects those objects from a dataset D that are not dominated by any others. An object p having d attributes (dimensions) dominates an object q , if p is strictly better than q in at least one dimension and not worse than q in all other dimensions, for a defined comparison function. Without loss of generality, we consider subsets of \mathbb{R}^d in which we search for *Skylines* with respect to (abbr. w.r.t.) the natural order \leq in each dimension.

Example 1. *The most cited example on Skyline queries is the search for a hotel that is cheap and close to the beach. Unfortunately, these two goals are conflicting as the hotels near the beach tend to be more expensive. Table I presents a sample dataset with its visualization in Figure 1. Each hotel is*

represented as a point in the two-dimensional space of price and distance to the beach. Interesting are all hotels that are not worse than any other hotel in these both dimensions.

TABLE I. Sample dataset of hotels.

hotel	id	beach dist. (km)	price (€)	board
	p1	2.00	25	none
	p2	1.25	50	breakfast
	p3	0.75	75	half board
	p4	0.50	150	full board
	p5	0.25	225	full board
	p6	1.75	110	half board
	p7	1.10	120	breakfast
	p8	0.75	220	full board
	p9	1.60	165	half board
	p10	1.50	185	breakfast

*The hotels p_6, p_7, p_9, p_{10} are dominated by hotel p_3 . The hotel p_8 is dominated by p_4 , while the hotels p_1, p_2, p_3, p_4, p_5 are not dominated by any other hotels and build the *Skyline* \mathcal{S} . From the *Skyline*, one can now make the final decision, thereby weighing the personal preferences for price and distance.*

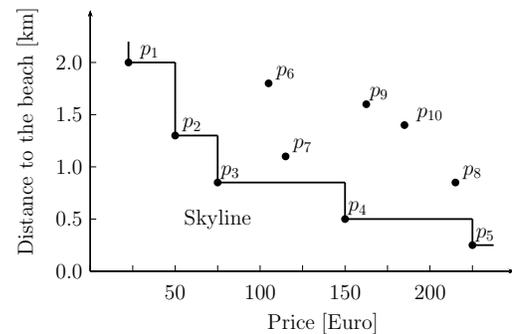


Figure 1. Skyline example.

Most of the work on *Skyline* computation has focused on the development of efficient algorithms for preference evaluation ([3] gives an overview). The most prominent algorithms are characterized by a tuple-to-tuple comparison-based approach [4][5]. Based on this, several algorithms have been published in the last decade, e.g., NN (Nearest Neighbor) [6], BBS (Branch and Bound Skyline) [7], SFS (Sort-Filter Skyline) [8], or LESS (Linear Elimination-Sort for Skyline) [9], just to name a few.

Unfortunately, the size of the Skyline \mathcal{S} can be very small (e.g., in low-dimensional spaces). Hence, a user might want to see the *next best objects behind the Skyline*.

Example 2. In our example above maybe five hotels are not enough, so we have to present the next stratum called \mathcal{S}_{ml}^1 (Skyline, multi-level 1, dashed line in Figure 2): p_6, p_7, p_8 . Also, the third best result set \mathcal{S}_{ml}^2 might be of interest: p_9, p_{10} .

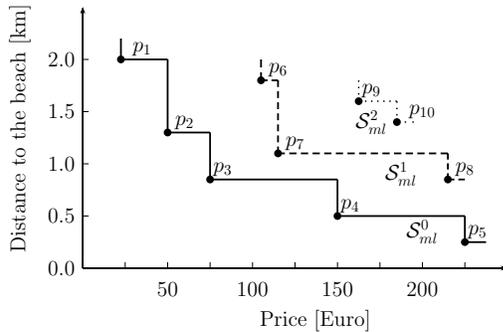


Figure 2. Multi-level Skylines.

Furthermore, in the presence of high-dimensional Skyline spaces, the size of the Skyline \mathcal{S} can still be very large, making it unfeasible for users to process this set of objects [3]. Hence, a user might want to see the *top-k* objects. That means a maximum of k objects out of the complete Skyline set if $|\mathcal{S}| \geq k$, or, for $|\mathcal{S}| < k$, use the Skyline set plus the next best objects such that there will be k results. In the previous example a *top-3* Skyline query would identify, e.g., p_1, p_2 , and p_3 , whereas in a *top-10* query it is necessary to consider the second and third stratum to identify p_6, p_7, p_8, p_9 , and p_{10} as additional Skyline points.

In this work, we propose evaluation strategies for *multi-level and top-k Skyline* queries, which do not depend on tuple comparisons. For this we generalize the well-known Skyline queries to *multi-level Skylines* \mathcal{S}_{ml} . We present an efficient algorithm to compute the l -th stratum of a Skyline query exploiting the lattice structure constructed over low-cardinality domains. Following [3][10][11], many Skyline applications involve domains with small cardinalities – these cardinalities are either inherently small (such as star ratings for hotels), or can naturally be mapped to low-cardinality domains (such as price ranges on hotels). In addition, we propose an evaluation strategy for *top-k Skyline* queries, which is based on the multi-level approach.

This paper is an extended version of [1] and additionally contains a deeper background on lattice-based Skyline computation, additional theoretical results, detailed description of the multi-level and top-k Skyline algorithms, examples, more comprehensive experiments, and extended related work. In addition we provide a section about an external implementation of our algorithms such that they do not rely on large main memory.

The remainder of this paper is organized as follows: In Section II we present the formal background. Based on this background we will discuss *multi-level Skyline* computation in Section III and *top-k Skyline* computation in Section IV. In Section V we present an external version of our algorithm. Section VI contains some remarks. We conduct an extensive

performance evaluation on synthetic and real datasets in Section VII. Section VIII contains related work, and Section IX concludes our paper.

II. SKYLINE QUERIES REVISITED

In this section, we revisit the problem of Skyline computation and shortly describe the Lattice Skyline approach, since this is the basis of our algorithms.

A. Skyline Queries

The aim of a Skyline query is to find the *best objects* in a dataset D , i.e., $\mathcal{S}(D)$. More formally:

Definition 1 (Dominance and Indifference). Assume a set of vectors $D \subseteq \mathbb{R}^d$. Given $x = (x_1, \dots, x_d)$, $y = (y_1, \dots, y_d) \in D$, x dominates y on D , denoted as $x <_{\otimes} y$, if the following holds:

$$x <_{\otimes} y \iff \forall j \in \{1, \dots, d\} : x_j \leq y_j \wedge \exists i \in \{1, \dots, d\} : x_i < y_i \quad (1)$$

We call x and y indifferent on D , denoted as $x \sim y$ if and only if $\neg(x <_{\otimes} y) \wedge \neg(y <_{\otimes} x)$.

Note that following Definition 1 we consider subsets of \mathbb{R}^d in that we search for the Skyline w.r.t. the natural order \leq in each dimension. Equation (1) is also known as *Pareto ordering* [12][13][14][15].

Definition 2 (Skyline \mathcal{S}). The Skyline $\mathcal{S}(D)$ of D is defined by the maxima in D according to the ordering $<_{\otimes}$, or explicitly by the set

$$\mathcal{S}(D) = \{t \in D \mid \nexists u \in D : u <_{\otimes} t\} \quad (2)$$

In this sense we prefer the minimal values in each domain and write $x <_{\otimes} y$ if x is better than y .

Note that an extension to arbitrary orders specified by utility functions is obvious and that Skylines are not restricted to numerical domains [16]. For any universe Ω and orderings $<_i \in (\Omega \times \Omega)$ ($i \in \{1, \dots, d\}$) the Skyline w.r.t. $<_i$ can be computed, if there exist scoring functions $g_i : \Omega \rightarrow \mathbb{R}$ for all $i \in \{1, \dots, d\}$ such that $x <_i y \iff g_i(x) < g_i(y)$. Then the Skyline of a set $M \subseteq \Omega$ w.r.t. $(<_i)_{i=1, \dots, d}$ is equivalent to the Skyline of $\{(g(x_1), \dots, g(x_d)) \mid x \in M\}$.

In general, algorithms of the block-nested-loop class (BNL) [4] are probably the best known algorithms for computing Skylines. They are characterized by a tuple-to-tuple comparison-based approach, hence having a worst case complexity of $\mathcal{O}(n^2)$, and a best case complexity of the order $\mathcal{O}(n)$; n being the number of input tuples, cf. [9]. The major advantage of a BNL-style algorithm is its simplicity and suitability for computing the maxima of arbitrary partial orders. Furthermore, a multitude of optimization techniques [8][9] and parallel variants [17][18][19][20] have been developed in the last decade.

B. Lattice Skyline Revisited

Lattice-based algorithms depend on the lattice structure constructed by a Skyline query over low-cardinality domains. An attribute domain $\text{dom}(S)$ is said to be *low-cardinality* if its value is drawn from a set $S = \{s_1, \dots, s_m\}$, such that the set cardinality m is small. Examples for such algorithms are

Lattice Skyline [11] and Hexagon [10], both having a worst case linear time complexity. Both algorithms follow the same idea: the partial order imposed by a Skyline query constitutes a lattice.

Definition 3 (Lattice [21]). A partially ordered set D with operator ' $<_{\otimes}$ ' is a lattice if $\forall x, y \in D$, the set $\{x, y\}$ has a least upper bound and a greatest lower bound in D . If a least upper bound and a greatest lower bound is defined for all subsets of D , we have a complete lattice.

The proof that a Skyline query constitutes a lattice can be found in [21].

Visualization of such lattices is often done using *Better-Than-Graphs (BTG)* [22], graphs in which edges state dominance. The nodes in the BTG represent *equivalence classes*. Each equivalence class contains the objects mapped to the same feature vector of the Skyline query. All values in the same equivalence class are indifferent and considered substitutable.

Before we consider an example, we need the notion of $\max(A)$.

Definition 4 ($\max(A)$). $\max(A)$ is the maximum value for a preference on the attribute A .

Example 3. An example of a BTG over a 2-dimensional space is shown in Figure 3 where $[0..2] \times [0..4]$ describes a domain of integers where attribute $A_1 \in \{0, 1, 2\}$ and $A_2 \in \{0, 1, 2, 3, 4\}$ (abbr. $[2; 4]$, $\max(A_1) = 2$, $\max(A_2) = 4$). The arrows show the dominance relationship between elements of the lattice. The node $(0, 0)$ presents the best node, i.e., the least upper bound, whereas $(2, 4)$ is the worst node. The bold numbers next to each node are unique identifiers (ID) for each node in the lattice, cp. [10]. Nodes having the same level in the BTG are indifferent, i.e., for example, that neither the objects in the node $(0, 4)$ are better than the objects in $(1, 3)$ nor vice versa. A dataset D does not necessarily contain tuples for each lattice node. In Figure 3, the gray nodes are occupied (non-empty) with elements from the dataset whereas the white nodes have no element (empty).

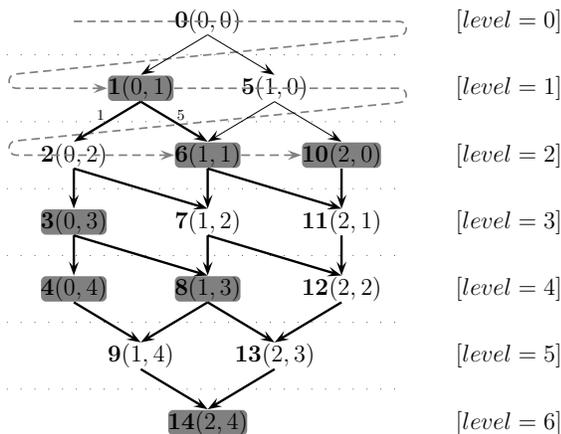


Figure 3. Lattice over $[0..2] \times [0..4]$.

The method to obtain the Skyline can be visualized using the BTG. The elements of the dataset D that compose the Skyline are those in the BTG that have no path leading to them from another non-empty node in D . In Figure 3, these

are the nodes $(0, 1)$ and $(2, 0)$. All other nodes have direct or transitive edges from these both nodes, and therefore are dominated.

When considering lattice algorithms the question arises how to map tuples t from a dataset D to the lattice structure. In [11] the authors use a function $F(t)$, which denotes a one-to-one mapping of an element $t \in D$ to a position in the BTG. For example, in the boolean case one can use the binary value of the boolean attributes to determine the array position, i.e., if $d = 3$, then the element $(true, false, true) \in D$ is represented by position 5 (101 in binary representation) in the BTG. A more general approach is presented in [10], where the position of a tuple is computed as below and also serves as the *unique identifier (ID)* mentioned in Example 3.

Lemma 1 (Edge Weights and Unique Node IDs). Let $S(D)$ be a Skyline query over a d -dimensional low-cardinality domain $\text{dom}(A) := \text{dom}(A_1) \times \dots \times \text{dom}(A_d)$, and $a = (a_1, \dots, a_d) \in \text{dom}(A)$.

a) The weight of an edge in the BTG expressing dominance between two direct connected nodes w.r.t. any attribute A_i is characterized by

$$\text{weight}(A_i) := \prod_{j=i+1}^d (\max(A_j) + 1) \quad (3)$$

For $j > d$ we set $\text{weight}(A_i) = 1$.

b) The unique identifier (ID) for $a \in \text{dom}(A)$ is given by

$$\text{ID}(a) = \sum_{i=1}^d (\text{weight}(A_i) \cdot a_i) \quad (4)$$

The proof of Lemma 1 can be found in [10].

Example 4. Reconsider Example 3. The edge weights of the node $a := (a_1, a_2) = (0, 1)$ are $\text{weight}(a_1) = 4 + 1 = 5$ and $\text{weight}(a_2) = 1$ as annotated in Figure 3. Hence, the unique identifier is $\text{ID}(a) = 5 \cdot 0 + 1 \cdot 1 = 1$, the bold number left of the node.

Note that the edge weights are also used to find the *direct dominated* nodes of a given node a . Just add the different edge weights to the unique identifier and one will get all direct dominated nodes, e.g., $1+1 = 2$ and $1 + 5 = 6$, both are dominated by node **1** in Figure 3. The pseudocode to find all direct dominated nodes is depicted in Algorithm 1 and is straightforward.

Lattice based algorithms exploit these observations to find the Skyline of a dataset over the space of vectors drawn from low-cardinality domains and in general consist of three phases. The pseudocode can be found in Algorithm 2. For details we refer to [10] and [11].

- 1) **Phase 1:** The *Construction Phase* initializes the data structures. The lattice is represented by an *array* in main memory (line 3 in Algorithm 2). Each position in the array stands for one node ID in the lattice. Initially, all nodes of the lattice are marked as *empty* and *not dominated*.
- 2) **Phase 2:** In the *Adding Phase* the algorithm determines for each element $t \in D$ the unique ID and therefore the node of the lattice that corresponds to t . This node will be marked as *non-empty* (line 7).

Algorithm 1 getDirectDominatedNodesBy(a)

Input: Node a .
Output: List of immediate dominated nodes.

```

1: function GETDIRECTDOMINATEDNODESBY(a)
2:   nodes  $\leftarrow$  list() // empty list to store dominated nodes
3:   // loop over the  $A_i$ 
4:   for  $i \leftarrow 1, \dots, d$  do
5:     // check if there is an edge for  $A_i$ 
6:     domNode  $\leftarrow$  ID( $a$ ) + weight( $A_i$ )
7:     if domNode is valid, then
8:       // domNode is in the next level and inside the BTG
9:       nodes.add(domNode)
10:    end if
11:  end for
12:  return nodes
13: end function

```

3) **Phase 3:** After all tuples have been processed, in the *Removal Phase* dominated nodes are identified. The nodes of the lattice that are marked as *non-empty* and which are not reachable by the transitive dominance relationship from any other *non-empty* node represent the Skyline values. Nodes that are *non-empty* but are reachable by the dominance relationship are marked *dominated* to distinguish them from present Skyline values.

From an algorithmic point of view this is done by a combination of *breadth-first traversal* (BFT) and *depth-first traversal* (DFT). The nodes of the lattice are visited level-by-level in a BFT (the dashed line in Figure 3, line 10 in Algorithm 2). Each time a *non-empty* and *not dominated* node is found, a DFT will start to mark dominated nodes as *dominated* (lines 12 – 17). The DFT does not need to explore branches already marked as *dominated*. The BFT can stop after processing a whole level not containing *empty* nodes hence marking the end of Phase 3.

For example, the node (0, 1) in Figure 3 is not empty. The DFT recursively walks down and marks all dominated nodes as *dominated* (thick black arrows). After the BFT has finished, the *non-empty* and *not dominated* nodes (here (0, 1) and (2, 0)) contain the Skyline objects.

III. MULTI-LEVEL SKYLINE COMPUTATION

In some cases it is necessary to return not only the best tuples as in common Skyline computation, but also to retrieve tuples directly dominated by those of the Skyline set (the *second stratum*), i.e., the tuples *behind the Skyline*. Following this method transitively, the input is partitioned into multiple levels (*strata*) in a way resembling the elements' quality w.r.t. the search preferences. In this section, we introduce the concept of *multi-level Skylines* and present an algorithm for efficient computation of iterated preferences in linear time.

A. Background

We extend Definition 2 of the Skyline by a level value to form *multi-level Skyline* (\mathcal{S}_{ml}) sets.

Definition 5 (Multi-Level Skyline \mathcal{S}_{ml}). *The multi-level Skyline set of level l (i.e., the l -th stratum) for a dataset D is defined as*

$$\begin{aligned} \mathcal{S}_{ml}^0(D) &:= \mathcal{S}(D) \\ \mathcal{S}_{ml}^l &:= \mathcal{S}\left(D \setminus \bigcup_{i=0}^{l-1} \mathcal{S}_{ml}^i(D)\right) \end{aligned}$$

Algorithm 2 Lattice Skyline (cp. [10][11])

Input: Dataset D with n tuples over d low-cardinality attributes, Array BTG of size V , V_i is the cardinality of dimension i .
Output: Skyline points.

```

1: // Phase 1: Construction Phase
2: Let  $V$  be the number of entries in the lattice,  $V \leftarrow V_1 \cdot \dots \cdot V_d$ 
3: Let  $BTG$  be an array of size  $V$  holding the different designators empty, non-empty, dominated, initialized to empty
4: // Let ID( $t$ ) be the unique identifier of a tuple  $t \in D$  and the index position in  $BTG$ .
5: // Phase 2: Adding Phase
6: for all  $t \in D$  do
7:   Set  $BTG[\text{ID}(t)]$  to non-empty
8: end for
9: // Phase 3: Removal Phase
10: for  $i \leftarrow 0 \dots V$  in a BFT do
11:   // Find all direct dominated nodes
12:   for all  $g \in \text{getDirectDominatedNodesBy}(BTG[i])$  do
13:     if  $BTG[g] == (\text{non-empty or dominated})$  then
14:        $BTG[g] \leftarrow \text{dominated}$ 
15:       do a DFT to mark successors as dominated
16:     end if
17:   end for
18: end for
19: // Output Skyline
20: for all  $t \in D$  do
21:   if  $BTG[\text{ID}(t)] == \text{non-empty}$  then
22:     output  $t$  as a Skyline point
23:   end if
24: end for

```

Thereby $\mathcal{S}_{ml}^0(D)$ is identical to the standard Skyline $\mathcal{S}(D)$ from Definition 2, and $\mathcal{S}_{ml}^{l_{max}}$ denotes the *non-empty* set with the highest level.

Example 5. For example, the query $\mathcal{S}_{ml}^1(D)$ on our hotel sample dataset computes the set of “second-best” tuples in the dataset D , i.e., the second stratum consisting of the objects p_6, p_7 , and p_8 as depicted in Figure 2, dashed line. The query $\mathcal{S}_{ml}^2(D)$ returns p_9, p_{10} .

Therefore, by iterating the Skyline operator one can rank the tuples in a given relation instance. Before we present our algorithm for multi-level Skyline computation we prove some properties.

Lemma 2. *For each tuple t in a finite dataset D , there is exactly one \mathcal{S}_{ml}^l set it belongs to:*

$$\forall t \in D : (\exists! l : t \in \mathcal{S}_{ml}^l(D)) \quad (5)$$

Proof: A Skyline query on $D \neq \emptyset$ never yields an empty result, i.e., $\mathcal{S}(D) \neq \emptyset$. Starting at 0, for each $l = 0, 1, 2, \dots$ the input dataset diminishes as all selection results for smaller values of l are removed from the input. Since $D \setminus \bigcup_{i=0}^l \mathcal{S}_{ml}^i(D) \subset D \setminus \bigcup_{i=0}^{l-1} \mathcal{S}_{ml}^i(D)$ and $|D|$ is finite, there has to be some l_{max} for which the following holds:

$$\bigcup_{i=0}^{l_{max}-1} \mathcal{S}_{ml}^i(D) \subset D \wedge \bigcup_{i=0}^{l_{max}} \mathcal{S}_{ml}^i(D) = D$$

So each tuple in D belongs to exactly one $\mathcal{S}_{ml}^l(D)$. ■

Lemma 2 shows that all tuples in a dataset belong to a \mathcal{S}_{ml} set of some level. So a kind of order on D w.r.t. the Skyline query is induced.

Lemma 3. All elements of $\mathcal{S}_{ml}^l(D)$ are dominated by elements of $\mathcal{S}_{ml}^i(D)$ for all $i < l$:

$$\forall y \in \mathcal{S}_{ml}^l(D) : (\exists x \in \mathcal{S}_{ml}^i(D) : x <_{\otimes} y) \text{ if } i < l \quad (6)$$

Proof: Consider a tuple $y \in \mathcal{S}_{ml}^l(D)$ that is not dominated by any element of $\mathcal{S}_{ml}^i(D)$ for $i < l$. Following Definition 5, $y \in \mathcal{S}_{ml}^i(D)$. This is a contradiction. ■

For every Skyline query on $D \neq \emptyset$ there is at least a \mathcal{S}_{ml} set of level 0. If it is the only one, no tuple in D is worse than any other w.r.t. the preference. Just as well, it is possible that all tuples in D belong to \mathcal{S}_{ml} sets of different levels. The Skyline query then defines a total order on the elements of D .

For each node x in the BTG, we can determine the stratum l of the \mathcal{S}_{ml}^l set it belongs to. Of course, all tuples in one equivalence class (which is represented by one node) are elements of the same \mathcal{S}_{ml}^l set. To find the \mathcal{S}_{ml}^l set for each node, we start at level 0 at the top node of the better-than graph. All tuples belonging to the standard Skyline set have a \mathcal{S}_{ml}^0 set level of 0. As the following lemma will show, the level of each tuple is the highest level of all tuples dominating it, increased by one:

Lemma 4 (\mathcal{S}_{ml}^l set level for an object). For an object $t \in D$ (or the BTG node representing its level values), $l(t) : \text{dom}(A) \rightarrow \mathbb{N}_0$ can be computed as follows:

$$l(t) := \begin{cases} 0 & \iff t \in \mathcal{S}_{ml}^0(D) \\ 1 + \max(\{l(s) | s \in D \wedge t <_{\otimes} s\}) & \iff t \notin \mathcal{S}_{ml}^0(D) \end{cases}$$

Proof: This follows from Definition 5 and Lemma 3. ■

With these concepts, we are now able to adjust lattice-based Skyline algorithms to compute multi-level Skyline sets.

B. The Multi-Level Lattice Skyline Algorithm (MLLS)

We will now see how the lattice based Skyline algorithms described in Section II-B can be adjusted to support multi-level Skyline computation. We call this algorithm *Multi-Level Lattice Skyline (MLLS)*. The first two phases of the standard lattice algorithms, *construction* and *adding*, remain unchanged. Modifications have to be done solely in the *removal phase*. Actually, as dominated nodes are not “removed” anymore, the *removal phase* is replaced by a *node classification phase*, cp. Algorithm 3.

The *classification phase* uses the same breadth-first and depth-first traversal as the original lattice Skyline algorithms. We need the node states *empty* and *non-empty*. In addition, we need to store a temporary value tmp_{ml} for the level of the \mathcal{S}_{ml} set a node belongs to currently. When a node n is reached, we reset the tmp_{ml} values for the nodes v_1, v_2, \dots , that are directly dominated by n . The value $\text{tmp}_{ml}(v_i)$ for a node v_i is computed as follows:

$$\text{tmp}_{ml}(v_i) = \begin{cases} \max(\text{tmp}_{ml}(v_i), \text{tmp}_{ml}(n)) & \iff n \text{ is empty} \\ \max(\text{tmp}_{ml}(v_i), \text{tmp}_{ml}(n) + 1) & \iff n \text{ is not empty} \end{cases}$$

Algorithm 3 Multi-Level Skyline – Classification Phase

Input: Better-Than Graph (BTG)

Output: list of \mathcal{S}_{ml} sets

```

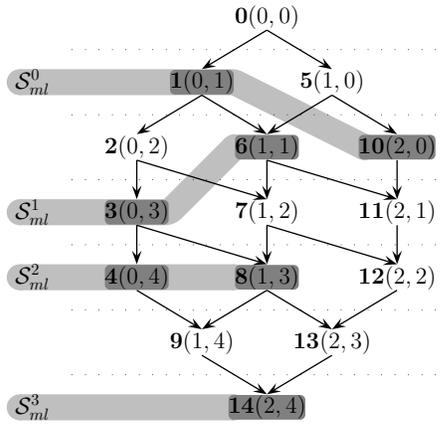
1: function CLASSIFY(BTG)
2:    $\mathcal{S}_{ml} \leftarrow \text{list}\langle \text{list}\rangle()$  // initialize list to store  $\mathcal{S}_{ml}$  sets
3:    $\text{tmp}_{ml}[\text{BTG}] \leftarrow 0$  // initialize  $\text{tmp}_{ml}$  array with 0's
4:   // iterate over all nodes  $n$  (BFT), start with node ID 0
5:    $n \leftarrow \text{node}(\text{ID} = 0)$ 
6:   repeat
7:     // use offset for  $\text{tmp}_{ml}$  computation
8:      $\text{offset} \leftarrow n.\text{isEmpty}() ? 0 : 1$ 
9:     // let  $\text{domNodes}$  be the list of direct dominated nodes
10:     $\text{domNodes} \leftarrow \text{getDirectDominatedNodesBy}(n)$ 
11:    for all  $v$  in  $\text{domNodes}$  do // compute  $\text{tmp}_{ml}$ 
12:       $\text{tmp}_{ml}(v) \leftarrow \max(\text{tmp}_{ml}(v), \text{tmp}_{ml}(n) + \text{offset})$ 
13:    end for
14:    // node not empty, add objects to  $\mathcal{S}_{ml}$  sets
15:    if ! $n.\text{isEmpty}()$  then
16:       $i \leftarrow \text{tmp}_{ml}[n]$ 
17:      // add all elements in node  $n$  to the  $\mathcal{S}_{ml}^i$  set
18:       $\mathcal{S}_{ml}.\text{addAll}(i, n.\text{getElements}())$ 
19:    end if
20:     $n \leftarrow \text{nextNode}()$  // next node in BFT
21:  until  $n == \text{NIL}$  // repeat until end of BTG is reached
22:  return  $\mathcal{S}_{ml}$ 
23: end function

```

In Algorithm 3, for a more convenient and efficient access to each of the \mathcal{S}_{ml} sets after the classification phase, we generate a list of nodes belonging to each \mathcal{S}_{ml} set while walking through the BTG. For this, we initialize a list of lists, which will store the \mathcal{S}_{ml}^i sets for each level i and an array of size of the BTG for the tmp_{ml} levels values (lines 2–3). Then we start the BFT at node 0 (line 5). If the current node n is empty we set an offset to 0, otherwise to 1 (line 8). The function `getDirectDominatedNodesBy()` retrieves all nodes directly dominated by n (cp. Algorithm 1). The complexity of this function is given by the number of Skyline dimensions as for each of them not more than one node can be dominated and we only visit directly dominated nodes (so the DFT ends at depth 1). The actual complexity of finding each of the directly dominated nodes or a node's successor in the BFT is specific to the representation of the BTG in memory, but can be assumed as $\mathcal{O}(1)$ [10][11]. For all direct dominated nodes compute the tmp_{ml} value in (lines 11–13). Afterward, if the node n contains elements from the input dataset, we retrieve for \mathcal{S}_{ml}^i the level i the elements belongs to (line 16) and add all elements of the node n to the \mathcal{S}_{ml}^i building up the multi-level Skyline sets (line 18). We continue with the next node in the BFT (line 20) until the end of the BTG is reached (line 21). The result is a list of \mathcal{S}_{ml}^i sets.

Example 6. Figure 4 visualizes an example of Algorithm 3.

Since node 0 is empty, the first relevant node is 1. Therefore we set $\text{tmp}_{ml}[1] = 0$ and add 1 to all direct dominated nodes, i.e., $\text{tmp}_{ml}[2] = \text{tmp}_{ml}[6] = 1$. We continue with node 5, which does not affect anything (the offset for the node is 0 and hence the tmp_{ml} values for the dominated nodes 6 and 10 remain unchanged). Since node 2 is empty, we set $\text{tmp}_{ml}[3] = \text{tmp}_{ml}[7] = 1$. Node 6 has already $\text{tmp}_{ml}[6] = 1$. The next node is 10, which still has $\text{tmp}_{ml}[10] = 0$. Node 3 sets $\text{tmp}_{ml}[4] = \text{tmp}_{ml}[8] = 2$, and so on. After the BFT has finished we have 4 \mathcal{S}_{ml} sets.

Figure 4. Multi-level Skyline S^l_{ml} .

Note that our multi-level Skyline algorithm has the same runtime complexity as the original Lattice Skyline algorithms [10][11]. In MLLS the construction and adding phase are unchanged in comparison to Lattice Skyline. The classification phase is a simple modification of the original removal phase without any additional overhead. Therefore, we state a linear runtime complexity of $\mathcal{O}(dV + dn)$, where d is the dimensionality, n is the number of input tuples, and V is the product of the cardinalities of the low-cardinality domains from which the attributes are drawn.

IV. TOP-K SKYLINE COMPUTATION

The concept of *top-k* ranking is used to rank tuples according to some score function and to return a maximum of k objects [23]. On the other hand, Skyline retrieves tuples where all criteria are equally important concerning some user preference [4]. However, the number of Skyline answers may be smaller than required by the user, for whom k are needed. Therefore, *top-k Skyline* was defined as a unified language to integrate them [24][25].

A. Background

Top-k Skyline allows to get exactly the top k from a partially ordered set stratified into subsets of non-dominated tuples. The idea is to partition the set into subsets (strata, multi-level Skyline sets) consisting of non-dominated tuples and to produce the top-k of these partitions.

In general, existing solutions calculate the first stratum with some sort of post-processing [24][25][26]. That means, after identifying the first stratum $S^0_{ml}(D)$, they remove the contained objects from the original input dataset D and continue Skyline computation on the reduced data. Hence, the second stratum is $S^1_{ml} = \mathcal{S}(D \setminus \mathcal{S}(D))$. This workflow is continued until k objects are found. Definition 6 outlines the three different cases which might occur:

Definition 6 (Top-k Skyline). A top-k Skyline query $\mathcal{S}^k_{tk}(D)$ on an input dataset D computes the top k elements with respect to the Skyline preferences. Formally:

- 1) If $|\mathcal{S}(D)| > k$, then return only k tuples from $\mathcal{S}(D)$, because not all elements can be returned due to result set size limitations. Any k tuples are a correct choice.

- 2) If $|\mathcal{S}(D)| = k$, then $\mathcal{S}^k_{tk}(D) = \mathcal{S}(D)$. That means return all tuples of $\mathcal{S}^0_{ml}(D)$. In this case there is no difference between the Skyline set and the top- k result set.
- 3) If $|\mathcal{S}(D)| < k$, then the elements of $\mathcal{S}(D)$ are not enough for an adequate answer. We have to find a value j , which meets the following criterion:

$$\left| \bigcup_{i=0}^{j-1} \mathcal{S}^i_{ml}(D) \right| < k \leq \left| \bigcup_{i=0}^j \mathcal{S}^i_{ml}(D) \right| \quad (7)$$

That means, not only all elements of $\mathcal{S}(D) = \mathcal{S}^0_{ml}(D)$ are returned, but also some of $\mathcal{S}^1_{ml}(D)$, and if the number of result tuples is still less than k , then $\mathcal{S}^2_{ml}(D)$, and so on. Note that from $\mathcal{S}^j_{ml}(D)$ exactly $k - \left| \bigcup_{i=0}^{j-1} \mathcal{S}^i_{ml}(D) \right|$ elements will be returned, which might not be all of it.

B. The Top-k Lattice Skyline Algorithm (TkLS)

In this section, we adapt the concept of *multi-level Skyline* computation in Section III to the computation of *top-k Skyline*.

Algorithm 3 returns a set of all \mathcal{S}^i_{ml} sets, hence the first k elements of these sets correspond to the *top-k* elements. However, in a top- k approach it is not necessary to compute all strata. To return the correct number of results, we will loop through the different \mathcal{S}^i_{ml} sets in order of their level and keep the sum of tuples belonging to them. We have to find the \mathcal{S}^i_{ml} sets that completely belong to the top- k results. That means, it is enough to compute l multi-levels such that

$$k \leq \left| \bigcup_{i=0}^l \mathcal{S}^i_{ml}(D) \right| \quad (8)$$

From an algorithmic point of view we adjust the multi-level Skyline algorithm as follows: Instead of dealing with node states like *dominated*, *non-empty*, or *empty* in the *Adding Phase* (cp. Section II-B), each node in the BTG is represented by an integer counter, counting the number of tuples belonging to the node. During the adding phase, this counter is increased. In addition, for each level of \mathcal{S}^i_{ml} , we keep track of the number of tuples belonging to it. Each time the \mathcal{S}^i_{ml} set level i for a non-empty node is determined, the number of tuples belonging to this \mathcal{S}^i_{ml} set is increased by the number of tuples belonging to the node. When all tuples are read, the classification is done just as described in Section III-B, Algorithm 3, but we append a simple break condition after line 19, which checks the above equation. Afterward, we can return the top- k elements.

C. Examples

In this section we provide some examples for top- k Skyline query computation and discuss different strategies for returning result objects.

Example 7. Consider a Skyline query on three attributes over the domain $[2; 2; 1]$. The lattice structure is given in Figure 5. The small index numbers next to each node show the number of tuples represented by each node. Nodes without index are empty.

After reading all input objects and classifying the nodes, k tuples should be returned. The different \mathcal{S}^i_{ml} set levels and their sizes can be found in Table II. We will see what happens for different values of k . The three cases correspond to those described in Definition 6.

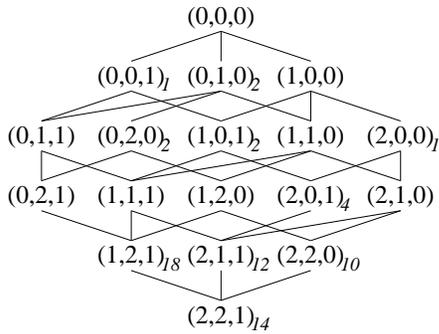


Figure 5. BTG for Example 7.

TABLE II. Nodes and S_{ml}^l set levels.

l	0	1	2	3	4
nodes in $S_{ml}^l(D)$	(0,0,1) (0,1,0) (2,0,0)	(0,2,0) (1,0,1)	(2,0,1) (1,2,1) (2,2,0)	(2,1,1)	(2,2,1)
$ S_{ml}^l(D) $	4	4	32	12	14

- 1) $k = 3$: Three of four tuples belonging to the nodes of $S_{ml}^0(D)$ are returned.
- 2) $k = 4$: $S_{ml}^0(D)$ is returned.
- 3) $k = 10$: $S_{ml}^0(D)$ and $S_{ml}^1(D)$ are returned completely, leading to 8 tuples in the result set. Additionally, $k - 8 = 2$ tuples from $S_{ml}^2(D)$ are returned.

Please note that the proposed algorithm will not return tuples in a progressive way. A tuple with a higher overall level than another could be returned, just because of the order of the input relation. The next example will outline such a scenario:

Example 8. Reconsider the Skyline query from Example 7. The top-1 result should be returned. Tuples belonging to the corresponding nodes are read in the following order:

$$(0, 2, 0), (2, 1, 1), (2, 0, 0), (0, 1, 0), \dots$$

The third tuple read is the first one in $S_{ml}^0(D)$. As the top-1 query is looking for only one result, the algorithm will stop after reading (and returning) $(2, 0, 0)$.

One may criticize that picking the top- k results from the different equivalence classes (nodes of the BTG) is arbitrary in some manner, especially if a multi-level set S_{ml}^j only partially belongs to the top- k result set. In most cases, there will be one S_{ml}^j set only partially belonging to the top- k results. From S_{ml}^j only $k - \left| \bigcup_{i=0}^{j-1} S_{ml}^i(D) \right|$ tuples have to be returned such that the total number of k results is matched. All other tuples are discarded. In this case we have to pick some arbitrary elements out of this S_{ml}^j set to fill up the top- k elements.

To handle this "problem" we can think about some kind of ordering or sorting before returning the top- k elements. Only those tuples belonging to some specific equivalence classes could be returned. The information on the number of tuples in the different equivalence classes may as well be another criteria. By using this information, a number of different top- k queries could be executed with only a small need for

computations. An additional weighting of the Skyline attributes can be used to sort the nodes differently. Still, information on the S_{ml}^l set level of a node can be used, taking it as some attribute result candidates are ordered by. Whichever additional conditions and characteristics are used, the top- k results can be taken then from the nodes coming first in the new order, as Example 9 shows.

Example 9. After the computations for answering the query in Example 7, some kind of presentation preference may induce a different weighting of the attributes. Imagine that the first attribute is more important than the second, and the second one is more important than the third. Such presentation preferences often are added to user preferences in online shops [27].

To answer this query, the set of non-empty nodes in the BTG has to be identified. Then, these nodes are ordered ascending w.r.t. the level value for the first attribute. Nodes with equal level value are ordered ascending w.r.t. their S_{ml}^l set level l . For the non-empty nodes of the BTG, this leads to the following order w.r.t. the query (with best elements being on top). Nodes pooled in a set $\{, \}$ remain unordered, due to Skylines being strict partial orders [12].

$$\begin{aligned} &\{(0,0,1), (0,1,0)\}, \\ &\quad \{(0,2,0)\}, \\ &\quad \{(1,0,1)\}, \\ &\quad \{(1,2,1)\}, \\ &\quad \{(2,0,0)\}, \\ &\{(2,0,1), (2,2,0)\}, \\ &\quad \{(2,1,1)\}, \\ &\quad \{(2,2,1)\} \end{aligned}$$

The nodes holding the top- k tuples than can now be identified by summing up the number of tuples belonging to each of the nodes. Using the values of Example 7 (and Figure 5), a top-5 query could be answered by returning all tuples of the nodes $(0, 0, 1)$, $(0, 1, 0)$, and $(0, 2, 0)$. Please note that a top-4 query would return all tuples of nodes $(0, 0, 1)$ and $(0, 1, 0)$, but not all of $(0, 2, 0)$.

We omit further discussions of the effects of different ordering strategies here as w.r.t. the original Skyline query, all candidates in S_{ml}^j are equally good results. We have seen that in spite of being developed for Skyline queries, our multi-level Skyline algorithm can easily be applied to top- k Skyline queries as well. Its greatest advantage remains: the linear runtime complexity in the number of input tuples and size of the BTG.

V. EXTERNAL LATTICE

All lattice-based algorithms (e.g., [10][11][20]) keep the complete lattice structure in main memory. Hence all nodes must be in memory, may it be in an array, a hash map, or some other data structure. Following [20], the memory requirements is linear w.r.t. the size of the BTG:

$$mem(BTG) = \left\lceil \frac{1}{4} \prod_{i=1}^m (\max(A_i) + 1) \right\rceil \quad (\text{in bytes})$$

where $\max(A_i)$ is the maximal value of attribute A_i in the low-cardinality domain $\text{dom}(A_1) \times \dots \times \text{dom}(A_m)$, cp. Definition 4.

However, memory requirements can be very high in some cases. The logical next step is to develop external algorithms for multi-level and top-k Skyline computation.

The idea behind our approach is to always keep one level of the BTG in memory. We need each node together with the information if a tuple belongs to it and its S_{ml}^i . For a BTG level c , we read through the input relation and mark each node we find a tuple belonging to it as *non-empty*. After all tuples have been read in the current round, we compute the next level $c+1$, with all nodes marked *empty*. We then walk through the nodes in level c and for each node n we set the S_{ml}^i of the dominated nodes to

$$\begin{aligned} S_{ml}^i & \text{ iff } n \text{ is empty and} \\ S_{ml}^{i+1} & \text{ iff } n \text{ is non-empty.} \end{aligned}$$

After this, the S_{ml}^i of level c can be written to external memory and removed from main memory. During this "information handover" to the next level, we need to keep two levels in memory. We need the level we just were working with to deliver information of domination to the next level. As we have to be able to deal with the BTG's levels with the most nodes, we require enough main memory to store the level with the most nodes twice. To analyze this amount, we need some information on the structure of the lattice (see [22]).

Theorem 1. For a BTG (lattice) over a d -dimensional Skyline query on the domain $\text{dom}(A) = \text{dom}(A_1) \times \dots \times \text{dom}(A_d)$ the following holds:

a) Height of the BTG:

$$\text{height}(BTG) = 1 + \sum_{i=1}^d \max(A_i) \quad (9)$$

b) Width of the BTG at a specific level l , $\text{width}(BTG, l)$:

$$\begin{aligned} \text{width}(BTG, l) & = w(l, \{A_1, \dots, A_d\}), \text{ where} \\ w(l, \{A_1, \dots, A_d\}) & = \sum_{i=0}^{\min(l, \max(A_1))} w(l-i, \{A_2, \dots, A_d\}), \\ & \text{if } d > 1 \end{aligned}$$

c) Maximum width of the BTG:

Due to the symmetrical structure of a BTG, the level with the maximum width will occur at level

$$\lfloor \frac{\text{height}(BTG)}{2} \rfloor$$

Note that an efficient algorithm for computing the width of the lattice for a given Skyline query can be found in [28].

As we have to work from level to level, it is very useful to be able to compute a *first node* for a given level l without any information about nodes in other levels. To create a given level we start with the left-most node. The level value for each A_j of the left-most node of a BTG for a Skyline query over $\text{dom}(A) := \text{dom}(A_1) \times \dots \times \text{dom}(A_d)$ can be found using the following expression:

$$\max(\min(l - \sum_{i=j+1}^d \max(A_i), \max(A_j)), 0) \quad (10)$$

With this formula we set the level values of A_d to the highest possible value for the given level l . Then we set the level value for A_{d-1} to $l - \max(A_d)$ (if possible). If $l - \max(A_d) > \max(A_{d-1})$, parts of the level sum l are used to "fill" the level value at position $d-2$ and so on.

Example 10. Consider Figure 3. The first node of level 3 for example is computed as follows:

- for A_1 : $\max(\min(3 - 4, 2), 0) = 0$
- for A_2 : $\max(\min(3, 4), 0) = 3$

Therefore, we have the node $(0, 3)$.

Algorithm 4 is a more generic version of this formula that can do the filling with only for just a part of the A_i in $\text{dom}(A)$. Applying Algorithm 4 to all A_i obviously yields equal results as above Equation (10) and produces the left-most node of a BTG w.r.t. the BFT.

Algorithm 4 FILL-UP

Input: node n , index, level l , Skyline query over $\text{dom}(A_1) \times \dots \times \text{dom}(A_d)$

Output: a BTG node

```

1: function FILL-UP(n, index, l)
2:   dist ← 0
3:   x ← n
4:   // loop over the  $A_i$ 
5:   for  $i \leftarrow d, \dots, \text{index}$  do
6:     // apply Equation (10)
7:      $x[i] := \max(\min(l - \sum_{j=i+1}^d x[j], \max(A_i)), 0)$ 
8:     dist ← dist + x[i]
9:   end for
10:  if dist ≠ 0 then
11:    // no valid distribution can be found
12:    return NIL
13:  end if
14:  return x
15: end function

```

To find the next node y in the BFT for a given node $x := (x_1, \dots, x_d)$, we shift one level value to a position more left (i.e., with a lower index i) and fill the rest of the node with Algorithm 4. This shift function is given in Algorithm 5. It is repeatedly used to create a whole level of the BTG in Algorithm 6. With all these helping procedures, we can define Algorithm 7, *External TkLS*.

Further optimizations are straightforward. As during the switch to the next level we have to keep two levels in memory, we can always keep at least two levels in memory and process tuples of all nodes of these. We actually are able to hold a higher number of levels in memory at most times. Starting at level 0, in the first step we can compute as many levels as we can fit into memory completely. Then the reading of the input relation can be done for multiple levels at once. The level switch then has to be adjusted to compute the S_{ml}^i of all nodes in memory, write those to external memory, and remove the information from main memory. Before the highest BTG

Algorithm 5 NEXT-NODE

Input: node n , Skyline query over $\text{dom}(A_1) \times \dots \times \text{dom}(A_d)$
Output: next node in same level in BFT search

```

1: function NEXT-NODE( $n$ )
2:    $l \leftarrow \sum_{i=1}^d n[i]$ 
3:   for  $i \leftarrow d, \dots, 1$  do
4:     if  $n[i] > 0$  then
5:        $n[i] = n[i] - 1$ 
6:       for  $j \leftarrow i - 1, \dots, 1$  do
7:         if  $n[j] < \max(A_j)$  then
8:            $n[j] = n[j] + 1$ 
9:            $\text{rem} \leftarrow l - \sum_{k=1}^j n[k]$ 
10:          return FILL-UP( $n, 1, l - \text{rem}$ )
11:        end if
12:      end for
13:    end if
14:  end for
15:  // no more node in current level can be found
16:  return NIL
17: end function

```

Algorithm 6 CREATE-LEVEL

Input: level l , Skyline query over $\text{dom}(A_1) \times \dots \times \text{dom}(A_d)$
Output: a list of nodes of the given level

```

1: function CREATE-LEVEL( $l$ )
2:   // init a list to hold nodes
3:   result  $\leftarrow$  list()
4:   node  $\leftarrow$  init node of level 0
5:   node  $\leftarrow$  FILL-UP(node,  $l$ )
6:   while (node  $\neq$  NIL) do
7:     result.add(node)
8:     node  $\leftarrow$  NEXT-NODE(node)
9:   end while
10:  // no more nodes in current level can be found
11:  return result
12: end function

```

level is processed, the next level has to be calculated - or more than one level, if they fit into memory. That way any amount of memory between the minimum requirement and enough to keep the whole BTG in main memory can be used efficiently. If more than two levels fit into memory in average, the number of loops through the input relation is lower than for one level in memory. The algorithm will actually turn into a variant of in-memory *TkLS* when the whole BTG fits into memory and the input relation has to read only once.

Please note that if working with more than one level at a time, we need to loop through the input relation twice for every set of levels in memory. In the first loop, we only mark non-empty nodes. After this first loop, we can do a BFT/DFT to find the correct S_{ml}^i for the nodes in memory. The second loop through the input relation will then return the tuples together with their S_{ml}^i . Tuples belonging to the lowest level in memory could be returned instantly in the first loop as for them the S_{ml}^i will not change anymore. As we work on at least two levels at a time, the number of loops through the input relation is at most as high as while working on only one level (i.e., $\max(A)$ loops). For the lowest (and highest) levels, potentially more than two levels fit into memory.

Algorithm 7 External TkLS

Input: Skyline query over $\text{dom}(A_1) \times \dots \times \text{dom}(A_d)$

```

1: prvLvl  $\leftarrow$  NIL
2: for  $i \leftarrow 0, \dots, \sum_{i=1}^d \max(A_i)$  do
3:   currLvl  $\leftarrow$  CREATE-LEVEL( $i$ )
4:   // walk through prvLvl and set  $S_{ml}^i$  for
5:   // all nodes in currLvl
6:   // read input relation and
7:   // • mark non-empty nodes in currLvl
8:   // • return each input tuple with its  $S_{ml}^i$ 
9:   prvLvl  $\leftarrow$  currLvl
10: end for

```

Memory requirements and worst case performance of our algorithms with an external BTG can be found in Lemma 5.

Lemma 5 (Properties of External TkLS).

a) The memory requirements for *TkLS* with an external lattice is given by:

$$\text{mem}(BTG) := 2 \cdot \max(\text{width}(BTG)) \quad (11)$$

b) The number of rounds the input relation has to be read is given by:

$$\text{rounds} := n \cdot \text{height}(BTG) \quad (12)$$

c) The runtime complexity is given by the number of rounds, hence

$$\mathcal{O}(n \cdot \text{height}(BTG)) \quad (13)$$

We will see *External TkLS* at work in Example 11.

Example 11. We will apply *External TkLS* to a Skyline query and an input dataset D resembling the ones known from Example 6 and Figure 4. Our main memory is big enough to hold information of 6 nodes (twice the width of the BTG).

Round 1:

- *currLvl* is level 0 with node (0,0). The node is marked as belonging to S_{ml}^0 .
- Input relation is read, but no tuples are found.

Round 2:

- *currLvl* is level 1 with nodes (0,1) and (1,0). All nodes are marked as belonging to S_{ml}^0 .
- Input relation is read. Node (1,0) is non-empty. Tuples belonging to it are returned as part of S_{ml}^0 .

Round 3:

- *currLvl* is level 2 with nodes (0,2), (1,1), and (2,0). Nodes (0,2) and (1,1) are marked as S_{ml}^1 , nodes (0,2) as S_{ml}^0 .
- Input relation is read. Nodes (1,1) and (2,0) are non-empty. Tuples are returned as parts of S_{ml}^0 resp. S_{ml}^1 .

The algorithm continues until Round 7 (for level 6) in which tuples in S_{ml}^3 are returned.

This algorithm can easily be adjusted to compute a "normal" Skyline. We just have to omit the S_{ml}^i and only keep the status (*empty*, *non-empty*, or *dominated*) for each node. Only nodes belonging to *non-empty* nodes will be returned as Skyline results. An additional stop criteria could be added as well: The Skyline is found as soon as one level only holds *non-empty* and *dominated* nodes.

VI. REMARKS

In this section we will discuss some additional points on our lattice-based multi-level and top-k algorithms.

A. High-Cardinality Domains

The lattice based multi-level and top-k algorithms are restricted to low-cardinality domains. Since this is a huge restriction to Skyline computation, we want to adjust our algorithms to handle high-cardinality domains as well. One approach to handle large domains is suggested in [29]. The idea is to use a down-scaling of a high-cardinality domain to a small domain such that a lattice based algorithm can be used as some kind of pre-filtering, which eliminates objects before the final Skyline computation. Afterward, a BNL style algorithm is necessary for final evaluation.

This approach is in focus when handling large domains for multi-level and top-k Skyline computation. Instead of eliminating objects in the lattice-based pre-filtering phase, a similar approach as in Section IV might be used. The down-scaling [29] leads to several comparable objects in the scaled equivalence classes. An additional counter for each equivalence class as well as an overall counter for each level in the scaled lattice could serve as a break condition similar to our top-k Skyline algorithm. Afterward, a BNL-style top-k algorithm like EBNL or ESFS (cp. [25]) could do the final computation of the k best objects.

B. Parallelization

Since multi-core processors are going mainstream, one might also think about a parallel variant of TkLS. For the first two phases, the *Construction Phase* and the *Adding Phase* (cp. Section II-B) we can follow the approach of [20], which presents different parallel implementations of the lattice-based algorithms. They also show how to parallelize phase 3, the *Removal Phase*, in Lattice Skyline. However, since in TkLS the removal phase is replaced by a *Classification Phase*, the proposed parallelization does not apply anymore. The parallelization of the classification phase shown in Algorithm 3 is not straightforward, since line 12 – where the temporary value tmp_{ml} is computed – depends on some pre-computations, which can only be done following sequential execution. Therefore, parallelization of TkLS is restricted to phase 1 and phase 2. However, following [20], the adding phase takes the most computation time. Hence, we expect an enormous speed-up in multi-level and top-k Skyline computation when applying phase 2 in parallel.

VII. EXPERIMENTS

This section provides our benchmarks on synthetic and real data to reveal the performance of the outlined algorithms.

A. Benchmark Framework

The concept of MLLS is the basis of TkLS, hence the performance of our top-k approach reflects the power of our multi-level Skyline algorithm. And since there is no competitor for MLLS, we only compared our approach *TkLS* to the state-of-the-art algorithms in generic top-k Skyline computation, *Extended Block-Nested-Loop* (EBNL) and *Extended Sort-Filter-Skyline* (ESFS) [25]. EBNL is a variant of the standard BNL algorithm [4] with the modification that each computed stratum is removed from the dataset and the Skyline is computed again. ESFS is an extension of SFS [8] exploiting some kind of data pre-sorting. In the worst-case EBNL and ESFS have a time complexity of $\mathcal{O}(n^2)$, whereas all lattice based algorithms have linear runtime complexity. Note that there are other top-k Skyline algorithms (cp. Section VIII), but all of them exploit some index structure. Since our algorithm is generic for all kind of input data, we do not compare index based algorithms to our approach. Also note that we used the non-external version of our algorithm, because EBNL and ESFS are implemented as main memory algorithms.

All algorithms have been implemented in Java 7.0. TkLS follows the implementation details given in [20] and [10]. The experiments were performed on a machine running Debian Linux 7.1 equipped with an Intel Xeon 2.53 GHz processor. Our prototype is available as open source project on GitHub <https://github.com/endresma/TopKSkyline.git>.

For our synthetic datasets we used the data generator commonly used in Skyline research [4]. We generated *anti-correlated* (anti), *correlated* (corr), and *independent* (ind) distributions and varied (1) the data cardinality n , (2) the data dimensionality d , and (3) the top- k value. For the experiments on real-world data, we used the well-known *Zillow* and *NBA* datasets. All datasets were adapted for low-cardinality Skyline computation. The Zillow dataset crawled from www.zillow.com contains more than 2M entries about real estate in the United States. Each entry includes number of *bedrooms* and *bathrooms*, *living area* in sqm, and *age* of the building. The NBA dataset is a small 5-dimensional dataset containing 17264 tuples, where each entry records performance statistics for a NBA player. Following [30], NBA is a fairly correlated dataset. Both datasets serve as real-world applications, which require finding the Skyline on data with a low-cardinality domain.

B. Experimental Results

We now present our experimental results.

Figure 6 presents the behavior of our TkLS algorithm on a typical top-k Skyline query. We chose a 3-dimensional dataset because this is very common in Skyline or Pareto preference selection. The low-cardinality domain was constructed by [3; 4; 4], which might correspond to attributes like "board" (none, breakfast, half board, full board), "star ratings" (1*-5*), and "price ranges" (e.g., [40-80[, [80-120[, [120-160[, [160-200[, [200-240]) in a search for the best hotel. To keep the hotel example we chose $k = 5$ (a user wants only to see a few hotels) and restricted the input size to 50 to 300 objects.

We present the runtime on anti-correlated, correlated, and independent data. Figure 6a shows the result on anti-correlated data. TkLS clearly outperforms EBNL and ESFS even though TkLS has some overhead in constructing the lattice. ESFS and

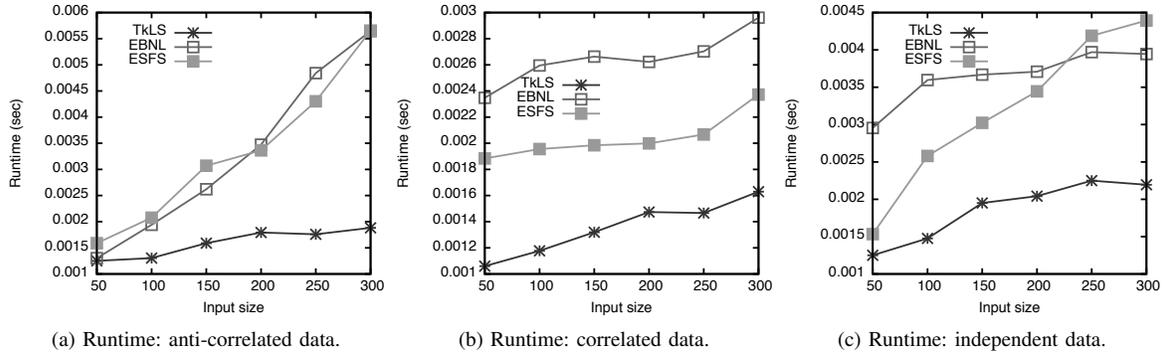


Figure 6. Runtime results on small input sizes: $d = 3$, top-5.

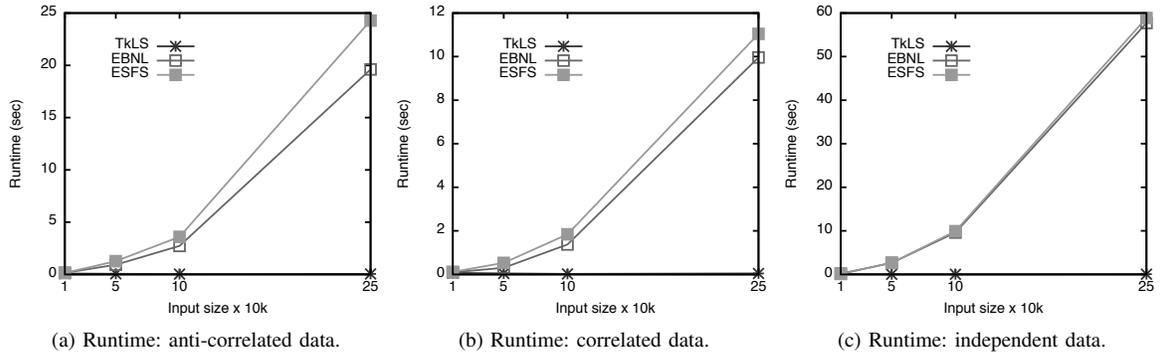


Figure 7. Runtime results on different input sizes: $d = 5$, top-500.

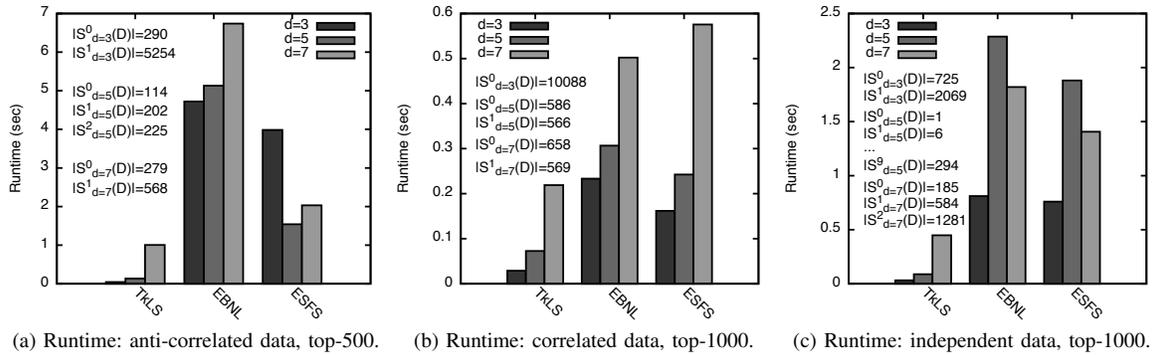


Figure 8. Experimental results on different dimensions: $d = 3, 5, 7$, $n = 50K$.

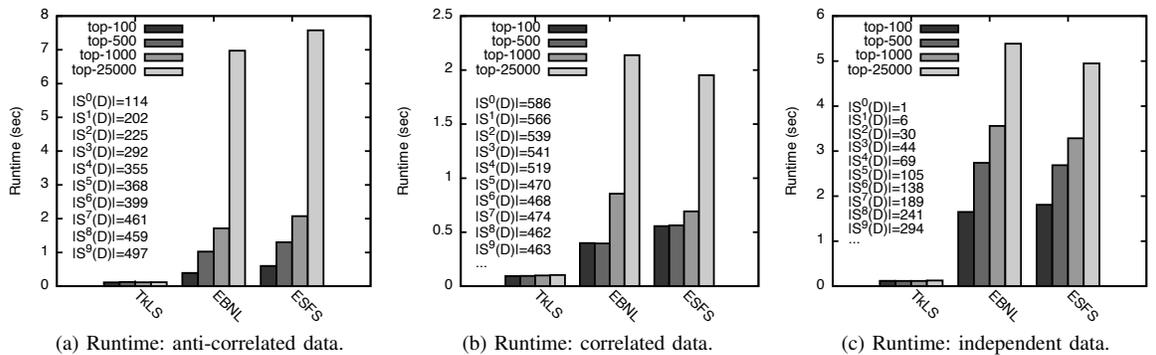


Figure 9. Influence of different k values: $k \in \{100, 500, 1K, 25K\}$, $d = 5$, $n = 50K$.

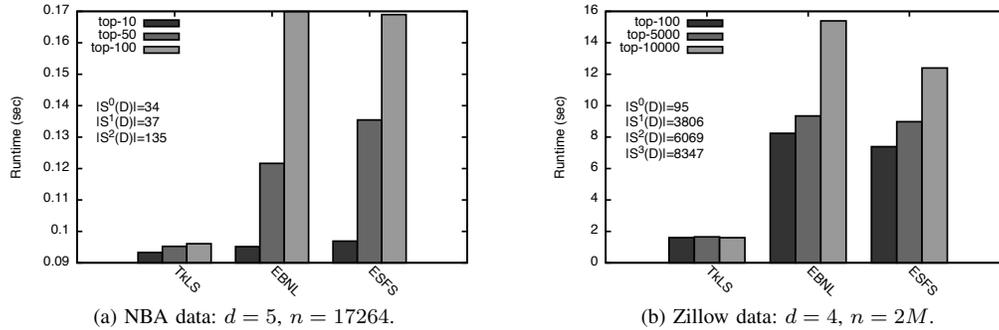


Figure 10. Experimental results on real data.

EBNL are almost equally good. Note that $|S_{ml}^0(D)| = 7$, that means the top-5 objects lie in the first stratum, i.e., the Skyline set. In the case of correlated (Figure 6b) and independent (Figure 6c) ESFS outperforms EBNL, but is worse than TkLS. Therefore, our multi-level algorithm is the first choice when evaluating Skyline queries on small datasets and low-cardinality domains.

Figure 7 presents the runtime of all algorithms on a 5-dimensional anti-correlated, correlated, and independent distributed dataset. We used a *top-500* query on different data cardinality. The low-cardinality domain was constructed by $[2; 3; 5; 10; 100]$. For all Skyline sets it holds that $|S(D)| < 500$ to get the effect of computing more than the 0-stratum. In contrast to the previous experiment we used larger input sizes (10K to 250K). Here, TkLS clearly outperforms EBNL and ESFS, which rely on a tuple-to-tuple comparison. TkLS always constructs the same lattice for all kind of data, and hence, has similar runtimes for all kind of input sizes.

Figure 8 shows the runtime of all algorithms on 3, 5, and 7 dimensions having anti-correlated, correlated, and independent data (up to $[2; 3; 5; 10; 10; 10; 100]$). The underlying data cardinality is $n = 50000$ and the target was to find the *top-500* respectively the *top-1000* elements. We also present the size of the different multi-level Skylines. For example, in the anti-correlated data (Figure 8a), if $d = 3$ we have $|S_{ml}^0(D)| = 290$ and the first stratum has $|S_{ml}^1(D)| = 5254$ objects. This is also the reason why ESFS in this case is worse than for $d = 5$ or $d = 7$. ESFS has to compare all objects of the first stratum to all others, not yet dominated tuples. Figure 8b and 8c show the results on correlated and independent datasets. In all our benchmarks TkLS outperforms EBNL and ESFS due its small lattice structure and independency of the data distribution, cp. [11].

Figure 9 visualizes the effect of different values of k for anti-correlated, correlated, and independent data distributions. Therefore we computed top- k elements for $k \in \{100, 500, 1K, 25K\}$ using 5 dimensions (as in Figure 7) and a data cardinality of $n = 50000$.

The runtime for TkLS for all k s and all distributions is very similar. This is due to the lattice based approach, where no tuple-to-tuple comparison is necessary, but only the construction of the BTG. Since the BTG for all k s is the same, the runtime for all top- k queries is quite similar.

Considering the results on anti-correlated data in Figure

9a, in the top-100 query only the Skyline $S(D)$ has to be computed. For $k = 500$ we have to compute stratum 0, 1, and 2. For top-1000 the first five strata are necessary, and for $k = 25K$ we need 41 strata to answer the query. We also see in this experiment that ESFS exploiting some pre-sorting is worse than EBNL. This is due the reordering of the elements. The results for correlated (Figure 9b) and independent data (Figure 9c) are quite similar. For the correlated and independent data we decided to use $k = 1000$ such that at least the second stratum must be computed to fulfill the 1000 objects.

Figure 10a shows the results on the NBA dataset, where the domain is drawn from $[10; 10; 10; 10; 10]$. The NBA set has a size of 17265 objects. Again, TkLS has similar runtime for each k whereas the runtime for EBNL and ESFS increases with larger k values.

In **Figure 10b** we present the Zillow dataset (domain $[10; 10; 36; 46]$). We compute the top- k elements for $k \in \{100, 5000, 10000\}$ to show the effect of computing different strata. TkLS clearly outperforms EBNL and ESFS. Again, since our algorithm is based on the lattice of a Skyline query the runtimes for the different k s are quite similar. EBNL and ESFS have to compute four strata to fulfill the $k = 10000$ query, which results in a long runtime and hence bad performance.

VIII. RELATED WORK

The most prominent algorithms for Skyline computation are based on a block-nested-loop style tuple-to-tuple comparison (e.g., BNL [4]). Based on this several algorithms have been published, e.g., the NN algorithm [6], SFS [8], or LESS [9], to just list some. Many of these algorithms have been adapted for parallel Skyline computation, e.g., [17][19][31]. There are also algorithms utilizing an index structure to compute the Skyline, e.g., [7][32]. Another approach exploits the lattice structure induced by a Skyline query over low-cardinality domains. Instead of direct comparisons of tuples, a lattice structure represents the better-than relationships. Examples for such algorithms are *Lattice Skyline* [11] and *Hexagon* [10], both having a *linear time complexity*. There is also work on parallel preference computation exploiting the lattice structure [20]. The authors of [29] present how to handle high-cardinality domains and therefore makes lattice algorithms available for a broad scope of applications.

The present paper is an extended version of [1], where the basics of multi-level and top- k Skylines were discussed.

The idea of *multi-level Skylines* was already mentioned by Chomicki [12] under the name of *iterated preferences*. However, Chomicki has never presented an algorithm for the computation of multi-level preferences.

In [33] the term “multi-level Skyline” was used, but it has nothing to do with the objects behind the Skyline. It was only used for a Skyline algorithm where a pre-computed Skyline dataset was used to compute continuous Skylines. Apart from that there is no other work on computing the i -th stratum of a Skyline query.

Regarding top-k [23][34] and Skyline [3] queries, there are some approaches that combine these both paradigms to *top-k Skyline queries*. In [24] and [25] the authors calculate the first stratum of the *Skyline* with some sort of post-processing. Afterward, they define the k best objects or continue Skyline computation without the first stratum. A similar approach was followed in [35], which presented the first study for processing top-k dominating queries over distance-based dynamic attribute vectors defined over a metric space. They present the Skyline-Based Algorithm (SBA) to compute the top-1 dominating object, which is removed from the dataset and the same process is repeated until all top-k results have been reported. The authors of [26] abstract Skyline ranking as a dynamic search over Skyline subspaces guided by user-specific preferences. In [36][37][38] and [39] an index based approach is used for top-k Skyline computation. However, index based algorithms in general cannot be used if there is a join or Cartesian product involved in the query. Su et al. [40] consider top-k combinatorial Skyline queries, and Zhang et al. [41] discuss a probabilistic top-k Skyline operator over uncertain data. Top-k queries are also of interest in the computation of spatial preferences [42][43], where the aim is to retrieve the k best objects in a spatial neighborhood of a feature object. Yu et al. [44] consider the problem of processing a large number of continuous top-k queries, each with its own preference. The authors of [45] present a framework for top-k query processing in large-scale P2P networks, where the dataset is horizontally distributed to peers. For this they compute k -skyband sets as a pre-processing step, which are aggregated to answer any incoming top-k queries.

Although there is some related work, the problem of efficiently evaluating top-k Skylines is still an open issue.

IX. CONCLUSION AND FUTURE WORK

In this paper we discussed the iterated evaluation of a Skyline query. For this, we provided a deep insight into lattice-based Skyline algorithms, and afterward presented how to modify Lattice Skyline to get multi-level Skyline sets. After a running through a set of input tuples, we are able to return not only the Pareto frontier, but also the tuples that are directly dominated by them, and so on. Our approach supports multi-level and top-k Skyline computation without computing each stratum of the Skyline query individually or relying on the time-consuming tuple-by-tuple comparisons. Comprehensive experiments show the benefit of our approach in comparison to existing techniques. Furthermore, the external version of our algorithm allows the evaluation of top-k Skyline queries even when main memory is low. For future work we want to get rid of the restriction on low-cardinality domains. Therefore, we want to adjust our algorithms as suggested in Section VI-A. However, this could be a challenging task.

REFERENCES

- [1] M. Endres and T. Preisinger, “Behind the Skyline,” in Proceedings of DBKDA '15: The 7th International Conference on Advances in Databases, Knowledge, and Data Applications. IARIA, 2015, pp. 141–146.
- [2] K. Stefanidis, G. Koutrika, and E. Pitoura, “A Survey on Representation, Composition and Application of Preferences in Database Systems,” ACM Transaction on Database Systems, vol. 36, no. 4, 2011.
- [3] J. Chomicki, P. Ciaccia, and N. Meneghetti, “Skyline Queries, Front and Back,” Proceedings of SIGMOD '13: ACM SIGMOD International Conference on Management of Data, vol. 42, no. 3, 2013, pp. 6–18.
- [4] S. Börzsönyi, D. Kossmann, and K. Stocker, “The Skyline Operator,” in Proceedings of ICDE '01: International Conference on Data Engineering. Washington, DC, USA: IEEE, 2001, pp. 421–430.
- [5] S. Mandl, O. Kozachuk, M. Endres, and W. Kießling, “Preference Analytics in EXASolution,” in Proceedings of BTW '15: Datenbanksysteme für Business, Technologie und Web, 2015.
- [6] D. Kossmann, F. Ramsak, and S. Rost, “Shooting Stars in the Sky: An Online Algorithm for Skyline Queries,” in Proceedings of VLDB '02: 28th International Conference on Very Large Data Bases, pp. 275–286.
- [7] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An Optimal and Progressive Algorithm for Skyline Queries,” in Proceedings of SIGMOD '03: Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM, 2003, pp. 467–478.
- [8] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, “Skyline with Presorting,” in Proceedings of ICDE '03: International Conference on Data Engineering, 2003, pp. 717–816.
- [9] P. Godfrey, R. Shipley, and J. Gryz, “Algorithms and Analyses for Maximal Vector Computation,” The VLDB Journal, vol. 16, no. 1, 2007, pp. 5–28.
- [10] T. Preisinger and W. Kießling, “The Hexagon Algorithm for Evaluating Pareto Preference Queries,” in Proceedings of MPref '07: Multidisciplinary Workshop on Preference Handling, 2007.
- [11] M. Morse, J. M. Patel, and H. V. Jagadish, “Efficient Skyline Computation over Low-Cardinality Domains,” in Proceedings of VLDB '07: International Conference on Very Large Data Bases, 2007, pp. 267–278.
- [12] J. Chomicki, “Preference Formulas in Relational Queries,” in TODS '03: ACM Transactions on Database Systems, vol. 28, no. 4. New York, NY, USA: ACM Press, 2003, pp. 427–466.
- [13] W. Kießling, “Foundations of Preferences in Database Systems,” in Proceedings of VLDB '02: International Conference on Very Large Data Bases. Hong Kong, China: VLDB, 2002, pp. 311–322.
- [14] W. Kießling, M. Endres, and F. Wenzel, “The Preference SQL System - An Overview,” Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society, vol. 34, no. 2, 2011, pp. 11–18.
- [15] P. Fishburn, “Preference Structures and their Numerical Representation,” Theoretical Computer Science, vol. 217, no. 2, 1999, pp. 359–383.
- [16] M. Endres, “The Structure of Preference Orders,” in Proceedings of ADBIS '15: The 19th East European Conference in Advances in Databases and Information Systems. Springer, 2015.
- [17] J. Selke, C. Lofi, and W.-T. Balke, “Highly Scalable Multiprocessing Algorithms for Preference-Based Database Retrieval,” in Proceedings of DASFAA '10: International Conference on Database Systems for Advanced Applications, ser. LNCS, vol. 5982. Springer, 2010, pp. 246–260.
- [18] S. Park, T. Kim, J. Park, J. Kim, and H. Im, “Parallel Skyline Computation on Multicore Architectures,” in Proceedings of ICDE '09: International Conference on Data Engineering. Washington, DC, USA: IEEE Computer Society, 2009, pp. 760–771.
- [19] S. Liknes, A. Vlachou, C. Doulkeridis, and K. Nørsvåg, “APSkyline: Improved Skyline Computation for Multicore Architectures,” in Proceedings of DASFAA '14: International Conference on Database Systems for Advanced Applications, 2014, pp. 312–326.
- [20] M. Endres and W. Kießling, “High Parallel Skyline Computation over Low-Cardinality Domains,” in Proceedings of ADBIS '14: East European Conference in Advances in Databases and Information System. Springer, 2014, pp. 97–111.

- [21] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, 2nd ed. Cambridge, UK: Cambridge University Press, 2002.
- [22] T. Preisinger, W. Kießling, and M. Endres, "The BNL++ Algorithm for Evaluating Pareto Preference Queries," in *Proceedings of MPref '06: Multidisciplinary Workshop on Preference Handling*, pp. 114–121.
- [23] I. F. Ilyas, G. Beskales, and M. A. Soliman, "A Survey of Top-k Query Processing Techniques in Relational Database Systems," *ACM Comput. Surv.*, vol. 40, no. 4, Oct. 2008, pp. 11:1–11:58.
- [24] M. Goncalves and M.-E. Vidal, "Top-k Skyline: A Unified Approach," in *OTM Workshops, 2005*, pp. 790–799.
- [25] C. Brando, M. Goncalves, and V. González, "Evaluating Top-k Skyline Queries over Relational Databases," in *Proceedings of DEXA '07: 18th International Conference on Database and Expert Systems Applications*, ser. LNCS, vol. 4653. Springer, 2007, pp. 254–263.
- [26] J. Lee, G.-W. You, and S.-W. Hwang, "Personalized Top-k Skyline Queries in High-Dimensional Space," *Information Systems*, vol. 34, no. 1, Mar. 2009, pp. 45–61.
- [27] S. Fischer, W. Kießling, and T. Preisinger, "Preference based Quality Assessment and Presentation of Query Results," in *G. Bordogna, G. Paila: Flexible Databases Supporting Imprecision and Uncertainty*, vol. 203. Secaucus, NJ, USA: Springer, 2006, pp. 91–121.
- [28] R. Glück, D. Köppl, and G. Wirsching, "Computational Aspects of Ordered Integer Partition with Upper Bounds," in *Proceedings of SEA '13: 12th International Symposium on Experimental Algorithms, 2013*, pp. 79–90.
- [29] M. Endres, P. Rooks, and W. Kießling, "Scalagon: An Efficient Skyline Algorithm for all Seasons," in *Proceedings of DASFAA '15: 20th International Conference on Database Systems for Advanced Applications, 2015*, in press.
- [30] C. Y. Chang, H. V. Jagadish, K.-L. Tan, A. K. H. Tung, and Z. Zhang, "On High Dimensional Skylines," in *Proceedings of EDBT '06: International Conference on Extending Database Technology*, ser. LNCS, vol. 3896. Springer, 2006, pp. 478–495.
- [31] S. Chester, D. Sidlauskas, I. Assent, and K. S. Bøgh, "Scalable Parallelization of Skyline Computation for Multi-Core Processors," in *Proceedings of ICDE '15: International Conference on Data Engineering, 2015*, pp. 1083–1094.
- [32] K. Lee, B. Zheng, H. Li, and W.-C. Lee, "Approaching the Skyline in Z Order," in *Proceedings of VLDB '07: The 33rd International Conference on Very Large Data Bases. VLDB Endowment, 2007*, pp. 279–290.
- [33] E. El-Dawy, H. M. O. Mokhtar, and A. El-Bastawissy, "Multi-level Continuous Skyline Queries (MCSQ)," in *Proceedings of ICDKE '11: International Conference on Data and Knowledge Engineering, IEEE, IEEE, 2011*, pp. 36–40, n/a.
- [34] M. J. Carey and D. Kossmann, "On saying 'Enough already!' in SQL," *Proceedings of SIGMOD '97: The ACM SIGMOD International Conference on Management of Data*, vol. 26, no. 2, 1997, pp. 219–230.
- [35] E. Tiakas, G. Valkanas, A. N. Papadopoulos, and Y. Manolopoulos, "Metric-Based Top-k Dominating Queries," in *Proceedings of EDBT '14: The 17th International Conference on Extending Database Technology, 2014*, pp. 415–426.
- [36] Y. Tao, X. Xiao, and J. Pei, "Efficient Skyline and Top-k Retrieval in Subspaces," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, 2007, pp. 1072–1088.
- [37] M. L. Yiu and N. Mamoulis, "Efficient Processing of Top-k Dominating Queries on Multi-Dimensional Data," in *Proceedings of the 33rd International Conference on Very Large Data Bases, ser. VLDB '07. VLDB Endowment, 2007*, pp. 483–494.
- [38] M. Goncalves and M.-E. Vidal, "Reaching the Top of the Skyline: An Efficient Indexed Algorithm for Top-k Skyline Queries," in *Database and Expert Systems Applications*, ser. LNCS. Springer, 2009, vol. 5690, pp. 471–485.
- [39] P. Pan, Y. Sun, Q. Li, Z. Chen, and J. Bian, "The Top-k Skyline Query in Pervasive Computing Environments," in *Joint Conferences on Pervasive Computing (JCPC). IEEE, 2009*, pp. 335–338.
- [40] I.-F. Su, Y.-C. Chung, and C. Lee, "Top-k Combinatorial Skyline Queries," in *Database Systems for Advanced Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, Jan. 2010*, pp. 79–93.
- [41] Y. Zhang, W. Zhang, X. Lin, B. Jiang, and J. Pei, "Ranking Uncertain Sky: The Probabilistic Top-k Skyline Operator," *Information Systems*, vol. 36, no. 5, Jul. 2011, pp. 898–915.
- [42] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k Spatial Preference Queries," in *Proceedings of ICDE '07: 23rd International Conference on Data Engineering. IEEE, 2007*, pp. 1076–1085.
- [43] J. B. Rocha-Junior, A. Vlachou, C. Doulkeridis, and K. Nørnvåg, "Efficient Processing of Top-k Spatial Preference Queries," in *Proceedings of the VLDB Endowment*, vol. 4, no. 2, 2010, pp. 93–104.
- [44] A. Yu, P. K. Agarwal, and J. Yang, "Processing a Large Number of Continuous Preference Top-k Queries," in *Proceedings of SIGMOD '12: The ACM SIGMOD International Conference on Management of Data. ACM, 2012*, pp. 397–408.
- [45] A. Vlachou, C. Doulkeridis, K. Nørnvåg, and M. Vazirgiannis, "Skyline-based Peer-to-Peer Top-k Query Processing," in *Proceedings of ICDE '08: The 2008 IEEE 24th International Conference on Data Engineering. Washington, DC, USA: IEEE Computer Society, 2008*, pp. 1421–1423.

Efficient Selection of Representative Combinations of Objects from Large Database

Md. Anisuzzaman Siddique, Asif Zaman, and Yasuhiko Morimoto

Graduate School of Engineering, Hiroshima University

Higashi-Hiroshima, Japan

Email: anis_cst@yahoo.com, {d140094@, morimoto@mis.}hiroshima-u.ac.jp

Abstract—Many applications require us to select combinations of objects from a database. To select representative combinations is one of the important processes for analysing data in such applications. Skyline query, which retrieves a set of non-dominant objects, is known to be useful to select representative objects from a database. Analogically, skyline query for combinations is also useful. Hence, we consider a problem to select representative distinctive combinations, which we call “objectsets”, in a numerical database in this paper. We analyse the properties of skyline objectset computation and develop filtering conditions to avoid needless objectset enumerations as well as comparisons among them. We perform a set of experiments to testify the importance and scalability of our skyline objectset method. In addition, we confirm that those filtering strategies also work for skyline objectset query variant called skyband objectset query. Therefore, we propose another method to compute skyband objectset skyline result. Our experiments also confirm the effectiveness and scalability of skyband objectset skyline method.

Keywords—Dataset; Skyline queries; Objectsets; Dominance relationship.

I. INTRODUCTION

This work propose an algorithm called complete objectset skyline (CSS) to resolve the objectsets skyline query problem. This article is an extended version of [1].

To select representative objects in a database is important to understand the data. Assume that we have a hotel database. To analyse the database, we, first, take a look at representative records, for example, the cheapest one, the most popular one, the most convenient one and so on. Skyline query [2] and its variants are functions to find such representative objects from a numerical database. Given a m -dimensional dataset D , an object O is said to dominate another object O' if O is not worse than O' in any of the m dimensions and O is better than O' in at least one of the m dimensions. A skyline query retrieves a set of non dominate objects. Consider an example in the field of financial investment. In general, an investor tends to buy the stocks that can minimize cost and risk. Based on this general assumption, the target can be formalized as finding the skyline stocks with smaller costs and smaller risks. Figure 1(a) shows seven stocks records with their costs (a_1) and risks (a_2). In the list, the best choice for a client comes from the skyline, i.e., one of $\{O_1, O_2, O_3\}$ in general (see Figure 1(b)).

A key advantage of the skyline query is that it does not require a specific ranking function; its results only depend on the intrinsic characteristics of the data. Furthermore, the skyline does not relay on different scales at different dimensions. For example risk unit or cost unit in Figure 1 may be not same but it does not affect the skyline query result. However, the order of the dimensional projections of the objects is important. Skyline query has broad applications including product or restaurant recommendations [3], review evaluations with user ratings [4],

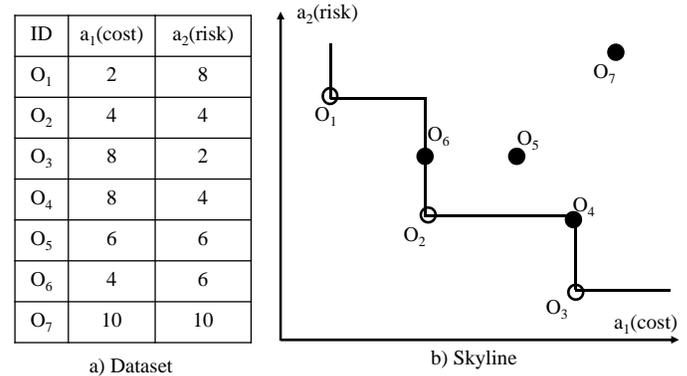


Figure 1. A skyline problem

querying wireless sensor networks [5], and graph analysis [6]. Algorithms for computing skyline objects have been discussed in the literature [7] [8] [9] [10].

One of the known limitations of the skyline query is that it can not answer various queries that require us to analyse not just individual object of a dataset but also their combinations. It is very likely that an investor has to invest more than one stock. For example, investment in O_1 will render the lowest cost. However, this investment is also very risky. Are there any other stocks or sets of stocks, which allow us to have a lower investment and/or a lower risk? These answers are often referred to as the investment portfolio. How to efficiently find such an investment portfolio is the principle issue studied in this work.

We consider a skyline query for distinctive combinations of objects (objectsets) in a database. Let k -objectset be a set, which contains k another object O' if O is not worse than O' in any of the m dimensions and O is better than O' in at least one of the m dimensions. A skyline query retrieves a set of non dominate objects. Consider an example in the field of financial investment. In general, an investor tends to buy the stocks that can minimize cost and risk. Based on this general assumption, the target can be formalized as finding the skyline stocks with smaller costs and smaller risks. Figure 1(a) shows seven stocks records with their costs (a_1) and risks (a_2). In the list, the best choice for a client comes from the skyline, i.e., one of $\{O_1, O_2, O_3\}$ in general (see Figure 1(b)). objects n be the number of objects in the dataset. The number of k -objectsets in the dataset amounts to ${}_n C_k$. We propose an efficient algorithm to compute variants of skyline query among the ${}_n C_k$ sets.

Assume an investor has to purchase two stocks. In Figure 1, conventional skyline query outputs $\{O_1, O_2, O_3\}$, which does not suggest sufficient information for the portfolio selection problem. Users may want to choose the portfolios, which

TABLE I. Sets of 2 Stocks

<i>ID</i>	$a_1(cost)$	$a_2(risk)$	<i>ID</i>	$a_1(cost)$	$a_2(risk)$	<i>ID</i>	$a_1(cost)$	$a_2(risk)$
$O_{1,2}$	6	12	$O_{2,4}$	12	8	$O_{3,7}$	18	12
$O_{1,3}$	10	10	$O_{2,5}$	10	10	$O_{4,5}$	14	10
$O_{1,4}$	10	12	$O_{2,6}$	8	10	$O_{4,6}$	12	10
$O_{1,5}$	8	14	$O_{2,7}$	14	14	$O_{4,7}$	18	14
$O_{1,6}$	6	14	$O_{3,4}$	16	6	$O_{5,6}$	10	12
$O_{1,7}$	12	18	$O_{3,5}$	14	8	$O_{5,7}$	16	16
$O_{2,3}$	12	6	$O_{3,6}$	12	8	$O_{6,7}$	14	16

are not dominated by any other sets in order to minimize the entire costs and risks. In such a case, if an user wants to select two stocks at a time from previous skyline result s/he can make two stock set such as $\{O_{1,2}, O_{1,3}, O_{2,3}\}$ and select any of them. However, set created from non dominant objects will be a non dominant set is not always true. For example, objectset $O_{1,3}$ is dominated by $O_{2,6}$ and there is no opportunity to judge this kind dominance relationship if we consider previous result only. That means an investor needs to create all two stocks sets after that perform domination check among those sets. It is very costly and not a very user friendly procedure. Table I shows sets consisting of two stocks, in which attribute values of each set are the sums of two component stocks. Objectsets $\{O_{1,2}, O_{2,3}, O_{2,6}\}$ cannot be dominated by any other objectsets (see Figure 2(a)) and, thus, they are the answers for the objectset skyline query. Furthermore, if the investor wants to buy three stocks then s/he needs to construct all of those three stocks sets and perform domination check among those sets to get the final result. In our running example, objectset skyline query for three stocks will retrieve objectsets $\{O_{1,2,3}, O_{1,2,6}, O_{2,3,4}, O_{2,3,6}\}$ as the query result.

Though a skyline query of objectsets is important in portfolio analysis, privacy aware data analysis, outlier-resistant data analysis, etc., there have been few studies on the objectsets skyline problem because of the difficulty of the problem. Su et al. proposed a solution to find the top- k optimal objectsets according to a user defined preference order of attributes [11]. However, it is hard to define a user preference beforehand for some complicated decision making tasks. Guo et al. proposed a pattern based pruning (PBP) algorithm to solve the objectsets skyline problem by indexing individuals objects [12]. The key disadvantage of the PBP algorithm is that it needs object selecting pattern in advance and the pruning capability depends on this pattern. Moreover, this algorithm is for fixed size objectset k and failed to retrieve result for all k .

We have introduced the objectsets skyline operator in 2010 [13]. In this work, we developed a method for finding the skyline objectsets that are on the convex hull enclosing all the objectsets. However, it misses objectsets that are not on the convex hull, which may provide meaningful results.

The main challenge in developing an method for objectset skyline is to overcome its computational complexity. The space complexity of objectset skyline computation is exponential in general and a dataset of n records can have up to $n C_k$ skyline objectsets. This means that the time complexity of objectset skyline is also exponential because we have to generate all skyline objectsets during the computation. Since the set of intermediate candidate objectsets may not fit in memory, conventional method have to generate all candidate objectsets

in a progressive manner and update the resultant objectset skyline dynamically. Thus, we cannot implement any index structures such as R-trees [9] and ZBTrees [14].

In this paper, we present an efficient solution that can select skyline objectsets, which include not only convex skyline objectsets but also non-convex skyline objectsets. The objectset size k can be varied from 1 to n and within which a user may select a smaller subset of his/her interest.

There is a well-known shortcoming in the skyline query. Though the result of the skyline query contains top-1 object for any scoring function, it cannot be used for selecting top- k ($k > 1$) object, which means that it cannot be used if an user wants more than one object for a specific scoring function. For example, if an user wants to choose top-3 cheapest stocks, the result of the skyline query does not contain all the top-3. Our previous objectset skyline query also has this shortcoming of the skyline query.

To solve the problem, in this paper, we also examine the objectset of another variant of skyline query called “skyband-objectset” query. Skyband-objectset query for K -skyband returns a set of objectsets, each objectset of which is not dominated by K other objectsets. In other words, an objectset in the skyband-objectset query may be dominated by at most $K - 1$ other objectsets.

The skyband-objectset query helps us to retrieve desired objectsets without any scoring function. It can increase (decrease) the number of objectsets by increasing (decreasing) the skyband value of K . From skyband-objectset result an user can easily choose his/her desired objectsets by applying top- k set queries. For the dataset in Figure 1, the skyband-set query for objectset size $s = 1$ and $K = 1$ retrieves all non dominated objectsets i.e., $\{O_1, O_2, O_3\}$. Again, from Figure 1(b) for $s = 1$ and $K = 2$ we get objectsets $\{O_1, O_2, O_3, O_6\}$ those are dominated by at most one objectset. Here O_4, O_5 , and O_7 are dominated by more than one objectset so they are excluded from the query result. Next, for $s = 2$ and $K = 1$ it retrieves objectsets $\{O_{1,2}, O_{2,3}, O_{2,6}\}$ shown by double circles in Figure 2(a). Similarly, for $s = 2$ and $K = 2$ skyband-set query will retrieve $\{O_{1,2}, O_{1,6}, O_{2,3}, O_{2,6}, O_{3,4}\}$ (see Figure 2(b)). In this result, objectsets $O_{1,6}$ and $O_{3,4}$ are included because of dominated by only one objectset. Figure 2(b) shows that objectset $O_{1,6}$ is dominated by objectset $O_{1,2}$ and objectset $O_{3,6}$ is dominated by objectset $O_{2,3}$. Therefore, one can use the skyband-set query at the preprocessing step for top- k set query. It is useful for candidate set generation to select his/her desired objectsets without any scoring function.

We introduce CSS to resolve the objectsets skyline query problem. It progressively filters the objectsets that are impossible to be the objectsets skyline result, and uses a filtering mech-

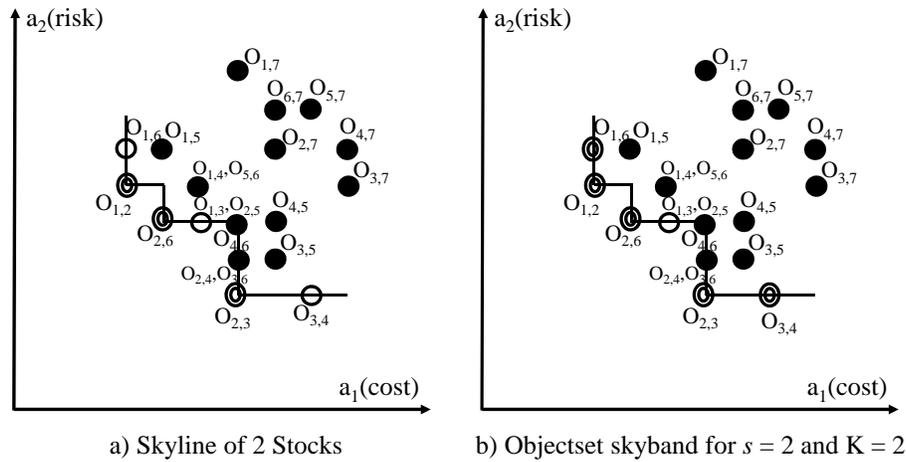


Figure 2. Objectset skyline problem

anism to retrieve the skyline objectsets without enumerating all objectsets. We develop two filtering techniques to avoid generating a large number of unpromising objectsets. Moreover, we confirm that proposing filtering strategies is also useful for skyband objectset query computation. We also propose another method called K -skyband objectset skyline (KSS) to compute skyband objectset query result. The efficiency of the both algorithms are then examined with experiments on a variety of synthetic and real datasets.

The remaining part of this paper is organized as follows. Section II reviews the related work. Section III discusses the notions and basic properties for skyline objectset as well as the problem of skyband objectset. In Section IV, we specify details of our algorithms with proper examples and analysis. We experimentally confirm our algorithms in Section VI under a variety of settings. Finally, Section VII concludes this work and describes our future intention.

II. RELATED WORK

Our work is motivated by previous studies of skyline query processing as well as objectsets skyline query processing. Therefore, we briefly review conventional skyline query and objectset skyline query proposed in this work.

A. Skyline Query Processing

Borzsonyi et al. first introduced the skyline operator over large databases and proposed three algorithms: *Block-Nested-Loops(BNL)*, *Divide-and-Conquer (D&C)*, and B-tree-based schemes [2]. BNL compares each object of the database with every other object, and reports it as a result only if any other object does not dominate it. A window W is allocated in main memory, and the input relation is sequentially scanned. In this way, a block of skyline objects is produced in every iteration. In case the window saturates, a temporary file is used to store objects that cannot be placed in W . This file is used as the input to the next pass. *D&C* divides the dataset into several partitions such that each partition can fit into memory. Skyline objects for each individual partition are then computed by a main-memory skyline algorithm. The final skyline is obtained by merging the skyline objects for each partition. Chomicki et al. improved BNL by presorting, they proposed *Sort-Filter-Skyline(SFS)* as a variant of

BNL [7]. Among index-based methods, Tan et al. proposed two progressive skyline computing methods *Bitmap* and *Index* [15]. In the *Bitmap* approach, every dimension value of a point is represented by a few bits. By applying bit-wise *AND* operation on these vectors, a given point can be checked if it is in the skyline without referring to other points. The index method organizes a set of m -dimensional objects into m lists such that an object O is assigned to list i if and only if its value at attribute i is the best among all attributes of O . Each list is indexed by a B-tree, and the skyline is computed by scanning the B-tree until an object that dominates the remaining entries in the B-trees is found. The current most efficient method is *Branch-and-Bound Skyline(BBS)*, proposed by Papadias et al., which is a progressive algorithm based on the *best-first nearest neighbor (BF-NN)* algorithm [9]. Instead of searching for nearest neighbor repeatedly, it directly prunes using the R^* -tree structure.

Recently, more aspects of skyline computation have been explored. Chan et al. proposed k -dominant skyline and developed efficient ways to compute it in high-dimensional space [16]. Lin et al. proposed n -of- N skyline query to support online query on data streams, i.e., to find the skyline of the set composed of the most recent n elements. In the cases where the datasets are very large and stored distributively, it is impossible to handle them in a centralized fashion [17]. Balke et al. first mined skyline in a distributed environment by partitioning the data vertically [18]. Vlachou et al. introduce the concept of extended skyline set, which contains all data elements that are necessary to answer a skyline query in any arbitrary subspace [19]. Tao et al. discuss skyline queries in arbitrary subspaces [20]. More skyline variants such as dynamic skyline [21] and reverse skyline [22] operators also have recently attracted considerable attention.

B. Objectsets Skyline Query Processing

There are two closely related works, which are “top- k combinatorial skyline queries” [11] and “convex skyline objectsets” [13]. Su et al. studied how to find top- k optimal combinations according to a given preference order in the attributes. Their solution is to retrieve non-dominant combinations incrementally with respect to the preference until the best k results have been found. This approach relies on the

preference order of attributes and the limited number (top- k) of combinations queried. Both the preference order and the top- k limitation may largely reduce the exponential search space for combinations. However, in our problem there is no preference order nor the top- k limitation. Consequently, their approach cannot solve our problem easily and efficiently. Additionally, in practice it is difficult for the system or a user to decide a reasonable preference order. This fact will narrow down the applications of [11]. Siddique and Morimoto studied the “convex skyline objectset” problem. It is known that the objects on the lower (upper) convex hull, denoted as CH , is a subset of the objects on the skyline, denoted as SKY . Every object in CH can minimize (maximize) a corresponding linear scoring function on attributes, while every object in SKY can minimize (maximize) a corresponding monotonic scoring function [2]. They aim at retrieving the objectsets in CH , however, we focus on retrieving the objectsets in $CH \subseteq SKY$. Since their approach relies on the properties of the convex hull, it cannot extend easily to solve complete skyline problem.

The similar related work is “Combination Skyline Queries” proposed in [12]. Guo et al. proposed a pattern based pruning (PBP) algorithm to solve the objectsets skyline problem by indexing individuals objects. The key problem of PBP algorithm is that it needs object selecting pattern in advance and the pruning capability depends on this pattern. For any initial wrong pattern this may increase the exponential search space. Moreover, it fails to vary the cardinality of objectset size k . Our solution does not require to construct any pattern previously and also vary the objectset size k from 1 to n . There are some other works focusing on the combination selection problem but related to our work weakly [23] [24]. Roy et al. studied how to select “maximal combinations”. A combination is “maximal” if it exceeds the specified constraint by adding any new object. Finally, the k most representative maximal combinations, which contain objects with high diversities, are presented to the user. Wan et al. study the problem to construct k profitable products from a set of new products that are not dominated by the products in the existing market [24]. They construct non-dominated products by assigning prices to the new products that are not given beforehand like the existing products. Moreover, there exist no previous work that focus on skyband objectset query and are not suitable to solve this type query.

III. PRELIMINARIES

This section formally defines objectset skyline query and objectset Skyband query and studies their basic properties.

Given a dataset D with m -attributes $\{a_1, a_2, \dots, a_m\}$ and n objects $\{O_1, O_2, \dots, O_n\}$. We use $O_i.a_j$ to denote the j -th dimension value of object O_i . Without loss of generality, we assume that smaller value in each attribute is better.

Dominance

An object $O_i \in D$ is said to dominate another object $O_j \in D$, denoted as $O_i \leq O_j$, if $O_i.a_r \leq O_j.a_r$ ($1 \leq r \leq m$) for all m attributes and $O_i.a_t < O_j.a_t$ ($1 \leq t \leq m$) for at least one attribute. We call such O_i as *dominant object* and such O_j as *dominated object* between O_i and O_j . For example, in Figure 1(b) object O_7 is dominated by object O_5 . Thus, for this relationship, object O_5 is considered as dominant object and O_7 as dominated object.

Skyline

An object $O_i \in D$ is said to be a *skyline object* of D , if and only if does not exist any object $O_j \in D$ ($j \neq i$) that dominates O_i , i.e., $O_j \leq O_i$ is not true. The skyline of D , denoted by $Sky(D)$, is the set of skyline objects in D . For dataset shown in Figure 1(a), object O_2 dominates $\{O_4, O_5, O_6, O_7\}$ and objects $\{O_1, O_3\}$ are not dominated by any other objects in D . Thus, skyline query will retrieve $Sky(D) = \{O_1, O_2, O_3\}$ (see Figure 1(b)).

In the following, we first introduce the concept of objectset, and then use it to define objectsets skyline. A k -objectset s is made up of k objects selected from D , i.e., $s = \{O_1, \dots, O_k\}$ and for simplicity denoted as $s = O_{1, \dots, k}$. Each attribute value of s is given by the formula below:

$$s.a_j = f_j(O_{1.a_j}, \dots, O_{k.a_j}), (1 \leq j \leq m) \quad (1)$$

where f_j is a monotonic aggregate function that takes k parameters and returns a single value. For the sake of simplicity, in this paper we consider that the monotonic scoring function returns the sum of these values, i.e.,

$$s.a_j = \sum_{i=1}^k O_i.a_j, (1 \leq j \leq m) \quad (2)$$

though our algorithm can be applied on any monotonic aggregate function. Recall that the number of k -objectsets in D is $nC_k = \frac{n!}{(n-k)!k!}$, we denote the number by $|S|$. If we consider stocks shown in Figure 1, then the total number of objectset for two stocks is $7C_2$ i.e., $|S| = 21$ and objectset $O_{1,2}$ is made up from object O_1 and O_2 .

Dominance Relationship

A k -objectset $s \in D$ is said to dominate another k -objectset $s' \in D$, denoted as $s \leq s'$, if $s.a_r \leq s'.a_r$ ($1 \leq r \leq m$) for all m attributes and $s.a_t < s'.a_t$ ($1 \leq t \leq m$) for at least one attribute. We call such s as *dominant k -objectset* and s' as *dominated k -objectset* between s and s' . For example, in Figure 2(a) objectset $O_{1,6}$ is dominated by objectset $O_{1,2}$. In this relationship objectset $O_{1,2}$ is considered as dominant objectset and $O_{1,6}$ as dominated objectset.

Objectsets Skyline

A k -objectset $s \in D$ is said to be a *skyline k -objectset* if s is not dominated by any other k -objectsets in D . The skyline of k -objectsets in D , denoted by $Sky_k(D)$, is the set of skyline k -objectsets in D . Assume $k = 2$, then for the dataset shown in Table III, 2-objectset $O_{1,2}$, $O_{2,3}$, and $O_{2,6}$ are not dominated by any other 2-objectsets in D . Thus, 2-objectset skyline query will retrieve $Sky_2(D) = \{O_{1,2}, O_{2,3}, O_{2,6}\}$ (see Figure 2(a)).

Domination Objectsets

Domination objectsets of k -objectsets, denoted by $DS_k(D)$ is said to be a set of all dominated k -objectsets in D . Since the 1-objectsets skyline result is $Sky_1(D) = \{O_1, O_2, O_3\}$, then the domination objectsets of 1-objectsets is $DS_1(D) = \{O_4, O_5, O_6, O_7\}$, i.e., $D - Sky_1(D)$.

TABLE II. domRelationTable for 1-objectsets

Object	Dominant Object
O_1	\emptyset
O_2	\emptyset
O_3	\emptyset
O_4	O_2, O_3
O_5	O_2, O_6
O_6	O_2
O_7	$O_1, O_2, O_3, O_4, O_5, O_6$

Objectset Skyband

Objectset skyband query returns a set of objectsets, each objectset of which is not dominated by K other objectsets. In other words, an objectset in the skyband-set query may be dominated by at most $K - 1$ other objectsets. If we want to apply skyband objectset query in D and choose objectset size $s = 2$ and $K = 2$, then the objectset skyband will retrieve $\{O_{1,2}, O_{1,6}, O_{2,3}, O_{2,6}, O_{3,4}\}$ as query result.

IV. COMPLETE SKYLINE OBJECTSETS ALGORITHM

In this section, we present our proposed method called Complete Skyline objectSets (CSS). It is a level-wise iterative algorithm. Initially, CSS computes conventional skyline, i.e., 1-objectsets skyline then 2-objectsets skyline, and so on, until k -objectsets skyline.

For $k = 1$, we can compute 1-objectsets skyline using any conventional algorithms. In this paper, we use *SFS* method proposed in [7] to compute 1-objectsets skyline and receive the following domination relation table called *domRelationTable*.

For the objectsets skyline query problem, the total number of objectsets is $|S| = {}_n C_k$ for a dataset D containing n objects when we select objectsets of size k . This poses serious algorithmic challenges compared with the traditional skyline problem. For example, Brute Force approach needs to calculate each objectset s and also needs to judge domination check among all $|S|$ objectsets. Fortunately, large parts of the computations can be avoided with our proposed CSS algorithm. As Table III illustrates, $|S| = 21$ possible combinations are generated from only seven objects when $k = 2$. Even for a small dataset with thousands of entries, the number of objectsets is prohibitively large. Thanks to the Theorem 1, which gives us opportunity to prune many non-promising objectsets.

Theorem 1. *If all k members of an objectset s are in $DS_{k-1}(D)$, where $DS_{k-1}(D) = DS_1(D) \cup \dots \cup DS_{k-1}(D)$, then objectsets $s \notin Sky_k(D)$.*

Proof: Assume $s = \{O_1, \dots, O_k\}$ and $s \in Sky_k(D)$. Since all members of s are in $DS_{k-1}(D)$, then there must be k distinct dominant objectsets for each member of s . Suppose $\{O'_1, \dots, O'_k\}$ are those k distinct objectsets and $\{O'_{1, \dots, k}\}$ construct an objectset s' . Now, it implies dominance relationship $s' \leq s$, which contradict initial assumption $s \in Sky_k(D)$. Hence, a non dominant k -objectsets contains at least one skyline objectset. ■

Dominance relation given in Table II retrieves $DS_1(D) = \{O_4, O_5, O_6, O_7\}$. By using Theorem 1 for $k = 2$, we can safely prune ${}_4 C_2 = 6$ objectsets such as $\{O_{4,5}, O_{4,6}, O_{4,7},$

$O_{5,6}, O_{5,7}, O_{6,7}\}$ from $Sky_2(D)$ computation, since these combinations do not have any member from $Sky_{k-1}(D)$. After pruning by this theorem the number of remaining objectsets, i.e., 15 (21-6), is still too large for our running example. To avoid large objectsets comparison, CSS applies the second pruning strategy as follows:

Theorem 2. *Suppose S_1, S_2 , and S_3 be the three objectsets in D . If objectset $S_1 \leq S_2$, then $S_1 \cup S_3 \leq S_2 \cup S_3$ is true.*

Proof: Given that S_1, S_2 , and S_3 are the three objectsets in D and $S_1 \leq S_2$ is true. Now, if we think another objectset S_3 as a constant and add it on both side then we will get the relationship $S_1 \cup S_3 \leq S_2 \cup S_3$. Thus if objectset $S_1 \leq S_2$ is true, then $S_1 \cup S_3 \leq S_2 \cup S_3$ is also true. ■

Theorem 2 gives us another opportunity to eliminate huge number objectsets. Table II shows that object O_4 is dominated by O_2 and O_3 ($O_2 \leq O_4$ and $O_3 \leq O_4$). By using Theorem 2, we get following dominance relations for 2-objectsets: $O_{1,2} \leq O_{1,4}$, $O_{2,3} \leq O_{2,4}$, $O_{2,3} \leq O_{3,4}$. Similarly, object O_5 is dominated by O_2 and O_6 ($O_2 \leq O_5$ and $O_6 \leq O_5$). From this relation, we can derive $O_{1,2} \leq O_{1,5}$, $O_{2,3} \leq O_{3,5}$, and $O_{2,6} \leq O_{2,5}$. From $O_2 \leq O_6$, we can derive $O_{1,2} \leq O_{1,6}$, $O_{2,3} \leq O_{3,6}$. Finally, from the last row of Table II we have the dominance relationship $\{O_1, O_2, O_3, O_4, O_5, O_6 \leq O_7\}$. Now if we use objects $\{O_1, O_2, O_3\}$ as common objects then we get following additional dominance relationship $\{O_{1,2} \leq O_{1,7}$, $O_{2,3} \leq O_{3,7}$, $O_{2,6} \leq O_{2,7}\}$. Thus, according to Theorem 2, we can safely prune more 11 objectsets such as $\{O_{1,4}, O_{2,4}, O_{3,4}, O_{1,5}, O_{2,5}, O_{3,5}, O_{1,6}, O_{3,6}, O_{1,7}, O_{2,7}, O_{3,7}\}$ for $Sky_2(D)$. Actually, CSS algorithm will compose remaining (15-11) = 4 objectsets such as $\{O_{1,2}, O_{1,3}, O_{2,3}, O_{2,6}\}$ and it needs to perform domination checks among them. After performing domination check it retrieves $\{O_{1,2}, O_{2,3}, O_{2,6}\}$ as $Sky_2(D)$ query result. However, during this procedure CSS also updates the dominance relation table for 2-objectsets as shown in Table III.

For $k = 2$ dominance relation Table III retrieves domination objectsets $DS_2(D) = \{O_{1,3}, O_{1,4}, O_{1,5}, O_{1,6}, O_{1,7}, O_{2,4}, O_{2,5}, O_{2,7}, O_{3,4}, O_{3,5}, O_{3,6}, O_{3,7}\}$. When $k = 3$, conventional skyline algorithm needs to check dominance relation among $|S| = 35$ (${}_7 C_3$) objectsets. In contrast to such conventional algorithms, our CSS algorithm does not compose those 3-objectsets if the distinct 3 objects are in $DS_1(D)$ or $DS_1(D) \cup DS_2(D)$. For $DS_1(D) = \{O_4, O_5, O_6, O_7\}$, CSS does not compute ${}_4 C_3 = 4$ objectsets. They are $\{O_{4,5,6}, O_{4,5,7}, O_{4,6,7}\}$, and $O_{5,6,7}$. For $DS_1(D) \cup DS_2(D)$, CSS pruned another 22 objectsets. These 22 objectsets are $\{O_{1,3,4}, O_{1,3,5}, O_{1,3,6}, O_{1,3,7}, O_{1,4,5}, O_{1,4,6}, O_{1,4,7}, O_{1,5,6}, O_{1,5,7}, O_{1,6,7}, O_{2,4,5}, O_{2,4,6}, O_{2,4,7}, O_{2,5,6}, O_{2,5,7}, O_{2,6,7}, O_{3,4,5}, O_{3,4,6}, O_{3,4,7}, O_{3,5,6}, O_{3,5,7}, O_{3,6,7}\}$. After using Theorem 1, the remaining objectset number is reduced to 9 (35-26). After applying Theorem 2, CSS does not need to compute another 5 objectsets such as $\{O_{1,2,4}, O_{1,2,5}, O_{1,2,7}, O_{2,3,5}\}$, and $O_{2,3,7}$. Finally, the proposed algorithm will compose only four objectsets $\{O_{1,2,3}, O_{1,2,6}, O_{2,3,4}, O_{2,3,6}\}$ and perform domination check among these four objectsets to obtain $Sky_3(D)$. After the domination check, since these objectsets are not dominated by each other thus, CSS retrieves $\{O_{1,2,3}, O_{1,2,6}, O_{2,3,4}, O_{2,3,6}\}$ as $Sky_3(D)$ result. CSS will continue similar iterative procedure for the rest of the k values to compute $Sky_k(D)$.

TABLE III. domRelationTable for 2-objectsets

Objectset	Dom. Objectset	Objectset	Dom. Objectset
$O_{1,2}$	\emptyset	$O_{3,4}$	$O_{2,3}$
$O_{1,3}$	\emptyset	$O_{3,5}$	$O_{2,3}, O_{3,6}$
$O_{1,4}$	$O_{1,2}, O_{1,3}$	$O_{3,6}$	$O_{2,3}$
$O_{1,5}$	$O_{1,2}, O_{1,6}$	$O_{3,7}$	$O_{1,3}, O_{2,3}, O_{3,4}, O_{3,5}, O_{3,6}$
$O_{1,6}$	$O_{1,2}$	$O_{4,5}$	$O_{2,5}, O_{3,5}, O_{2,4}, O_{4,6}$
$O_{1,7}$	$O_{1,2}, O_{1,3}, O_{1,4}, O_{1,5}, O_{1,6}$	$O_{4,6}$	$O_{2,6}, O_{3,6}, O_{2,4}$
$O_{2,3}$	\emptyset	$O_{4,7}$	$O_{1,4}, O_{2,4}, O_{3,4}, O_{4,5}, O_{4,6}, O_{2,7}, O_{3,7}$
$O_{2,4}$	$O_{2,3}$	$O_{5,6}$	$O_{2,6}, O_{2,5}$
$O_{2,5}$	$O_{2,6}$	$O_{5,7}$	$O_{1,5}, O_{2,5}, O_{3,5}, O_{4,5}, O_{5,6}, O_{2,7}, O_{6,7}$
$O_{2,6}$	\emptyset	$O_{6,7}$	$O_{1,6}, O_{2,6}, O_{3,6}, O_{4,6}, O_{5,6}, O_{2,7}$
$O_{2,7}$	$O_{1,2}, O_{2,3}, O_{2,4}, O_{2,5}, O_{2,6}$		

V. OBJECTSETS SKYBAND COMPUTATION

We adapt previous CSS method for objectset skyband query computation. We term this method as k -skyband objectset (KSS). Initially consider a skyband objectset query where objectset size $s = 1$ and skyband size $K = 1$ i.e., conventional skyline query. Then it retrieves $\{O_1, O_2, O_3\}$ as the objectset skyband query result. If we keep objectset size $s = 1$ and increase the skyband value K to 2, then objectset skyband result becomes $\{O_1, O_2, O_3, O_6\}$. Similarly, skyband objectset query for $s = 1$ and $K = 3$ retrieves $\{O_1, O_2, O_3, O_4, O_5, O_6\}$. Finally, for $s = 1$ and $K = 4$ KSS retrieves all objects as a result.

Similar to objectset skyline query problem if we select objectsets of size s , then the number of objectsets is $|S| = {}_n C_s$ for a dataset D containing n objects. Unfortunately, to compute skyband objectset we can not use Theorem 1. This is because, even if all k member of an objectset s are in $DS_k(D)$, it can be retrieved as the member of objectset skyband. However, to compute objectset skyband query efficiently, we introduce Theorem 3, which is useful to filter objectsets as well as to reduce the number of comparisons required domination check.

Theorem 3. *If an objectset, say s , is dominated by at least K other objectsets, then we do not need to compose objectsets that contain s for K -skyband-set computation.*

Proof: Let S_1, S_2, S_3 , and S_4 be objectsets in DS . Assume that objectset S_1 is dominated by two objectsets S_2 and S_3 ($S_2 \leq S_1$ and $S_3 \leq S_1$). Thanks to Theorem 2, we can say that if $S_2 \leq S_1$ is true, then for super objectset $S_2 \cup S_4 \leq S_1 \cup S_4$ is also true. Similarly, $S_3 \cup S_4 \leq S_1 \cup S_4$ is true. There exist at least two other objectset such as $S_2 \cup S_4$ and $S_3 \cup S_4$ that can dominate super objectset $S_1 \cup S_4$. It implies that an objectset is dominated by two objectsets, it cannot be an objectset of 2-skyband-set. Thus, it is proved that if an objectset is dominated by at least K other objectsets then we do not need to compose super objectsets that contain the dominated objectset for skyband-set computation. ■

From Table II, we can easily construct similar *domRelationTable* for objectset size $s = 2$ as shown in Table III by using Theorem 2 and Theorem 3. Dominance relation Table III retrieves candidates for objectset skyband queries when objectset size $s = 2$. For example, if an user specifies skyband objectset query for $s = 2$ and $K = 1$, then the proposed algorithm will retrieve candidate objectsets $\{O_{1,2}, O_{1,3}, O_{2,3}, O_{2,6}\}$ from Table III. Note that the proposed algorithm will compose only four objectsets and

perform domination check among them to obtain skyband objectset result $\{O_{1,2}, O_{2,3}, O_{2,6}\}$. Here, objectset $O_{1,3}$ is dominated by objectset $O_{2,6}$. Next, if the user concerns about skyband objectset query with $s = 2$ and $K = 2$, then the proposed algorithm will choose candidate objectsets $\{O_{1,2}, O_{1,3}, O_{1,6}, O_{2,3}, O_{2,4}, O_{2,5}, O_{2,6}, O_{3,4}, O_{3,6}\}$ and perform domination check among these objectsets. Finally, it retrieves $\{O_{1,2}, O_{1,6}, O_{2,3}, O_{2,6}, O_{3,4}\}$ as skyband-set query result. The dominance relation Table III retrieves candidate objectsets for any skyband objectset query when $s = 2$. The proposed method will continue similar iterative procedures to construct dominance relation table each time for larger value of s and ready to report objectset skyband queries result for any skyband value of K .

VI. PERFORMANCE EVALUATION

In the experimental section, we empirically evaluated the performance of our proposed CSS and KSS approaches. We do not compare our algorithms with the algorithm for computing top-k combinatorial skyline [11]. The reasons are twofold. First, RCA requires the user to determine the ranking of every dimension of the dataset, and its performance varies depending on the user preference. Second, while our algorithms compare only groups of the same size, RCA compares groups of different sizes as well. However, in order to measure their relative performance, we adapt SFS skyline algorithm to compute set skyline algorithm [7]. To make the comparison fair, we have excluded all the pre-processing cost of SFS method such as cost of objectset generation.

We conduct a set of investigations with different dimensionalities (m), data cardinalities (n), and objectset size (k) to judge the effectiveness and efficiency of proposed methods. All experiments were run on a computer with Intel Core i7 CPU 3.4GHz and 4 GB main memory running on Windows. We compiled the source codes under jdk 1.8. Each experiment is repeated five times and we report the average results for performance evaluation. The execution times in the graphs shown in this section are plotted in log scale.

A. Performance on Synthetic Datasets

We prepared synthetic datasets with three data distributions correlated, anti-correlated, and independent, which are used in [2]. The results are shown in Figures 3, 4, and 5. The sizes of resulting synthetic datasets are varied from 2.3k to 161.7k depending on the number of objects n .

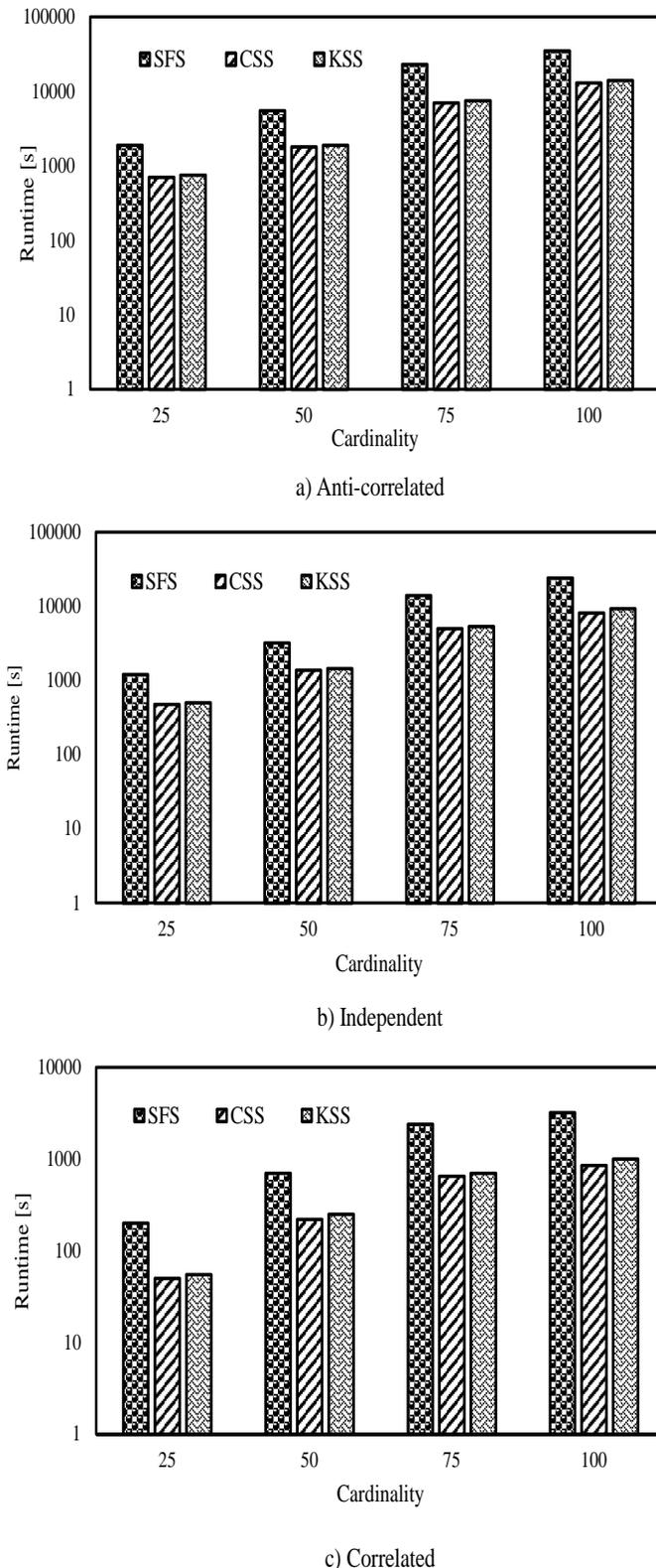


Figure 3. Performance for different cardinality

Effect of Cardinality

For this experiment, we specify the data dimensionality m to 4, objectset size k to 3, and vary dataset cardinality n . If the cardinality n takes the values of 25, 50, 75, and 100 then total objectset size become 2.3k, 19.6k, 67.5k, and 161.7k, respectively. We plot the running times of the algorithms in Figure 3. Figure 3(a), (b), and (c) respectively reports the performance on the correlated, the independent, and the anti-correlated datasets. The horizontal line represents the data cardinality and the vertical line represents execution time. We observe that all methods are affected by data cardinality. If the data cardinality increases then their performances fall down. The results demonstrate that proposed methods significantly outperforms the SFS method. The performance of SFS method degrades rapidly as the the dataset size increases, especially for the anti-correlated data distribution. This represents that the proposed methods can successfully prune the objectset composing as well as many unnecessary comparisons. However, the difference between CSS and KSS is not very significant. KSS needs little bit extra time to retrieve results for all K .

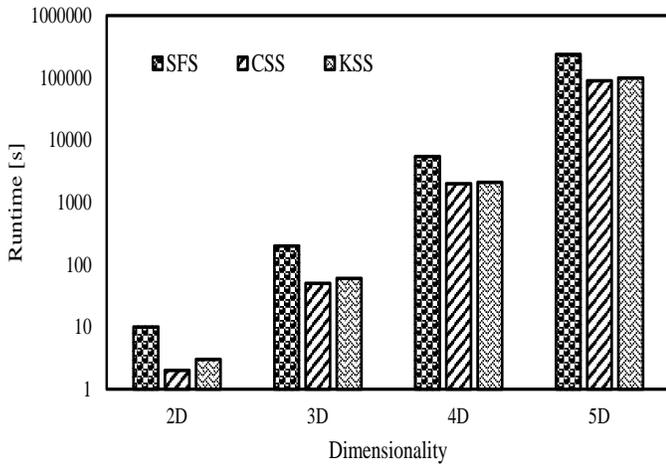
Effect of Dimensionality

In these experiments, we set the data cardinality n to 50, objectset size k to 3 and vary dataset dimensionality m ranges from 2 to 5. The elapsed time results are shown in Figure 4(a), (b), and (c). The horizontal line shows the data dimensionality m and the vertical line shows the execution time. The results showed that as the dimension m increases the performance of the all methods becomes slower. This is because checking the dominance relationship between two objectsets becomes more expensive with large values of m . For higher dimension, when the number of non dominant objectset increases the performance of all methods become sluggish. The running time of the proposed algorithms achieve satisfactory even when the dimension size is large. The results on correlated datasets are 9-20 times faster than the independent and the anti-correlated datasets. We observed that the time difference between CSS and KSS is not notable for the same reason as we discuss in previous section.

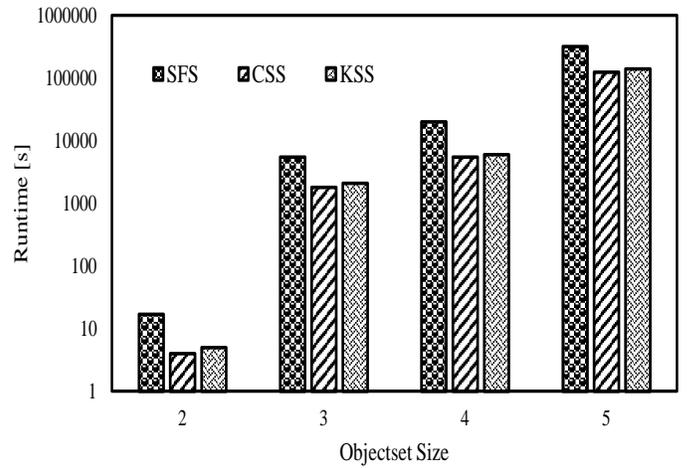
Effect of Objectset Size

In these experiments, we examine the performance of the proposed CSS and KSS under various objectset size k . We limit the data cardinality n to 50 and dataset dimensionality m to 4. The results are described in Figure 5(a), (b), and (c). The horizontal line shows the objectset size k and the vertical line shows the time taken. The results showed that as the objectset size k rises up, the performance of all methods are fallen down. The performance of SFS method is much worse than that of the proposed methods when the objectset size k is greater than 1. This is because the proposed methods use *SFS* algorithm for $k = 1$ to construct *domRelationTable*, and executing domination check. For higher value of s , it does not require to compose all objectset and it reduces the number of unnecessary comparisons. We notice that the time difference between CSS and KSS is not significant and CSS is faster that KSS.

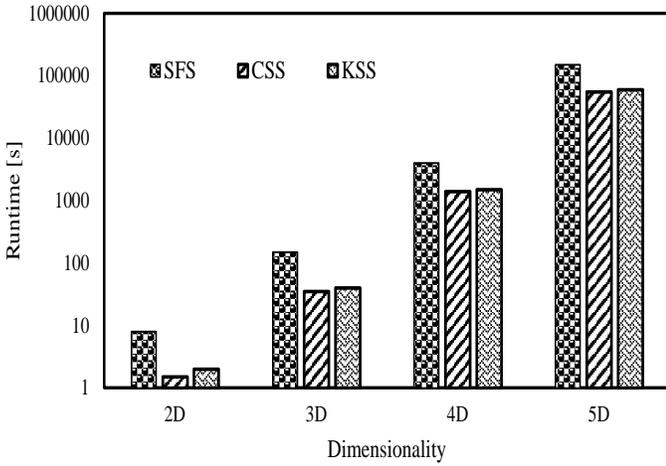
From the experimental results, we observe a pattern that the speed up of the proposed methods over SFS is 4-10 times faster. Since the number of skyline objectsets of the anti-correlated datasets is generally larger than those of the independent datasets and the correlated datasets, the algorithms



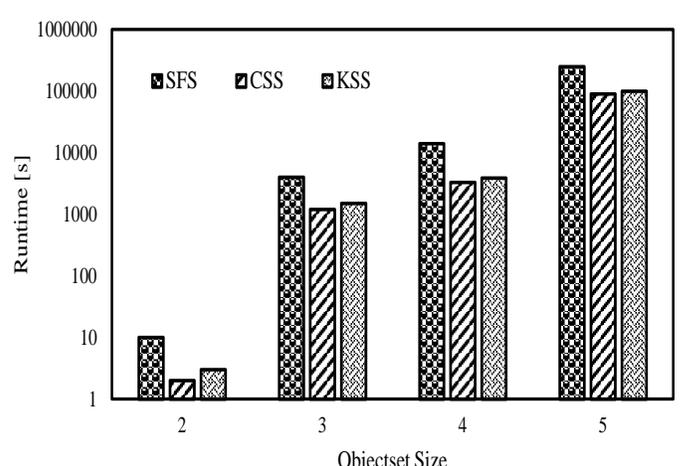
a) Anti-correlated



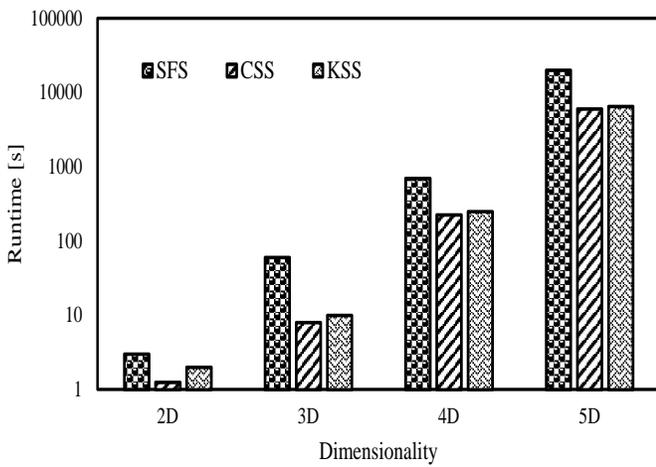
a) Anti-correlated



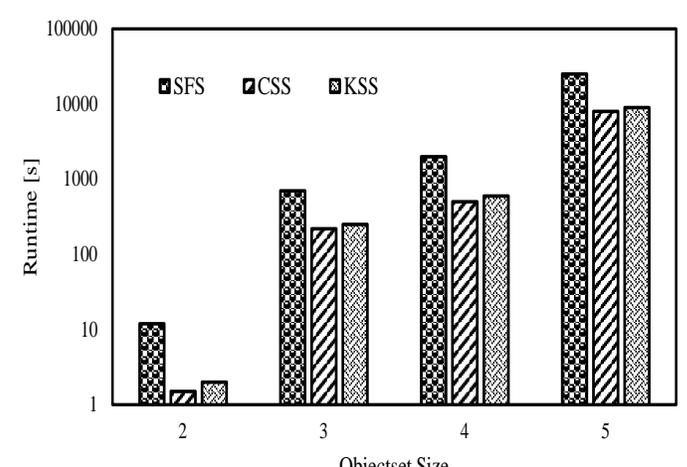
b) Independent



b) Independent



c) Correlated



c) Correlated

Figure 4. Performance for different data dimension

Figure 5. Performance for different objectset size

with the anti-correlated datasets take generally more execution time than those of the other datasets.

B. Performance on Real Dataset

We next present the experimental results of the proposed algorithms with real datasets. To examine the results for real dataset, we select the FUEL dataset, which is obtained from “www.fueleconomy.gov”. FUEL dataset is 24k 6-dimensional objects, in which each object stands for the performance of a vehicle (such as mileage per gallon of gasoline in city and highway, etc.). The attribute domain range for this dataset is [8, 89].

To deal with FUEL dataset, we conducted similar experiments like synthetic datasets. First, to study dimensionality, we specify the data cardinality n to 50, objectset size k to 3 and vary dataset dimensionality m from 2 to 5.

Figure 6(a) illustrates the performance the objectset skyline and objectset skyband queries of different dimension sizes. As the dimensions increases, the running time for all methods increases accordingly. However, CSS and KSS outperforms than SFS technique.

Our second experiments on real data examine the performance of different data cardinality n . For these experiments, we limit the dimensionality m to 4, objectset size k to 3, and vary dataset cardinality n from 25 to 100. The results are shown in Figure 6(b). As the data size increases, the running time for all methods increase sharply. We can observe that CSS is better than KSS and SFS.

In the final experiments, we examine the performance under various objectset size k . We fix the data cardinality n to 50 and dimensionality m to 4. The results are shown in Figure 6(c). As the objectset size increases, the execution time for all methods also increase. We can observe that the running time of CSS and KSS are much superior to SFS.

Notice that we get similar standardized results like independent distribution that represents the scalability of CSS and KSS on real dataset for all experiments with FUEL dataset.

These experiments demonstrate that our proposed CSS and KSS are consistently better than the SFS method on both synthetic and real datasets. Therefore, our experiments confirm the effectiveness and scalability of our algorithms.

VII. CONCLUSION

This paper addresses a skyline query for set of objects in a dataset. After analysis various properties of objectsets skyline, we propose an efficient and general algorithm called CSS to compute objectsets skyline. Because of the exponential growth in the number of combinations of tuples, objectsets skyline computation using conventional method is inherently expensive in terms of both time and space. Therefore, in order to prune the search space and improve the efficiency, we have developed two major pruning strategies. Using synthetic and real datasets, we demonstrate the scalability of proposed method. Intensive experiments confirm the effectiveness and superiority of our CSS algorithm.

In the future, first, we intend to implement the objectsets skyline problem when the aggregation function is not monotonic. As another direction for future work, we may consider a problem of finding a small number of representative skyline objectsets, similarly to finding a small number of

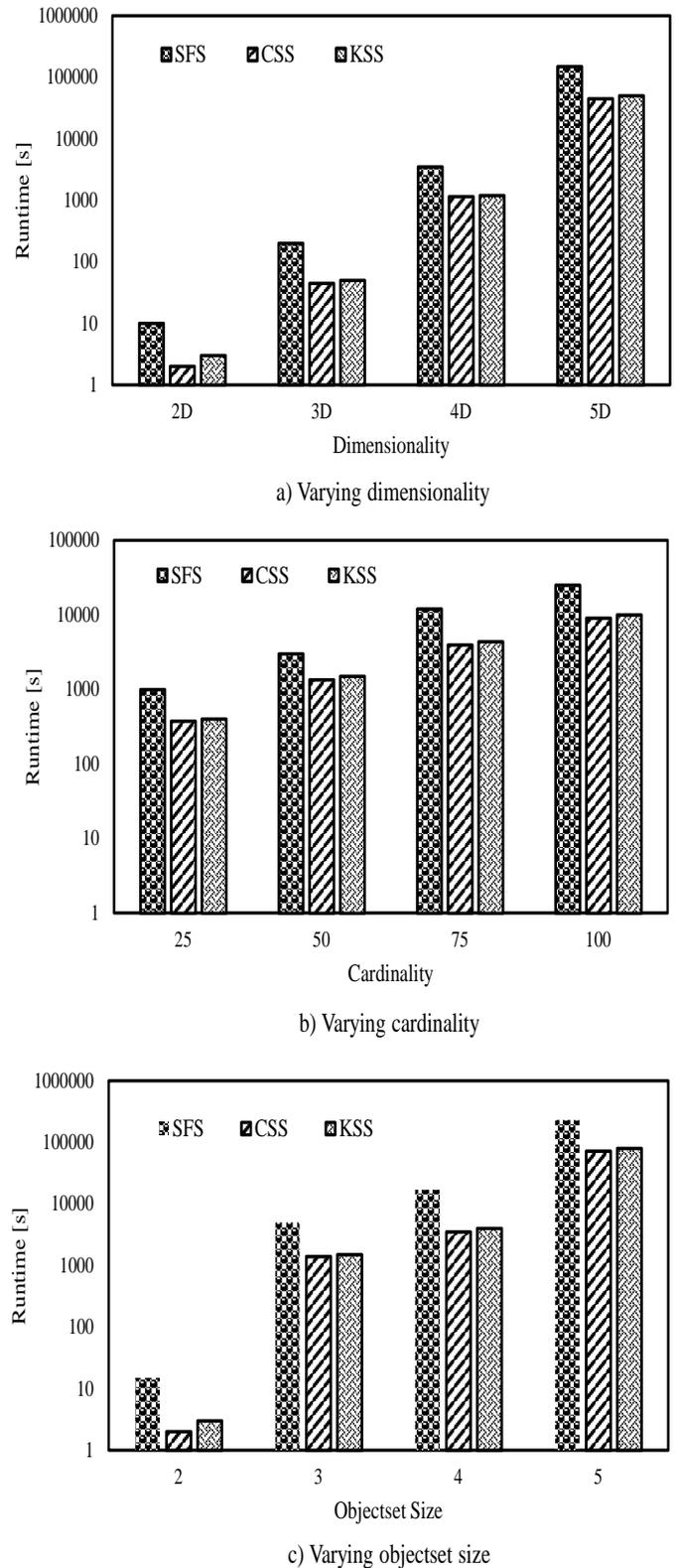


Figure 6. Experiments on FUEL dataset

representative skyline objectsets. Third, we want to design more optimized mechanisms for objectsets computation on distributed MapReduce environment.

ACKNOWLEDGMENTS

This work is supported by KAKENHI (23500180, 25.03040) Japan. M. A. Siddique is supported by Japanese government “JSPS postdoctoral fellowship for foreign researchers program” and A. Zaman is supported by Japanese government MEXT scholarship.

REFERENCES

- [1] M. A. Siddique, A. Zaman, and Y. Morimoto, “Skyline Objectset: Efficient Selection of Non-dominant Sets from Database,” in Proceedings of the 7th International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA) May 24–29, 2015, Rome, ITALY, 2015, pp. 114–120.
- [2] S. Borzsonyi, D. Kossmann, and K. Stocker, “The skyline operator,” in Proceedings of the 17th International Conference on Data Engineering (ICDE) April 2–6, 2001, Heidelberg, Germany, 2001, pp. 421–430.
- [3] J. Lee, S. Hwang, Z. Nie, and J.-R. Wen, “Navigation system for product search,” in Proceedings of the 26th International Conference on Data Engineering (ICDE) March 1–6, 2010, California, USA, 2010, pp. 1113–1116.
- [4] T. Lappas and D. Gunopulos, “CREST: Efficient confident search in large review corpora,” in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) September 20–24, 2010, Barcelona, Spain, 2010, pp. 467–478.
- [5] G. Wang, J. Xin, L. Chen, and Y. Liu, “Energy efficient reverse skyline query processing over wireless sensor networks,” *IEEE Transactions on Knowledge Data Engineering (TKDE)*, vol. 24, no. 7, 2012, pp. 1259–1275.
- [6] L. Zou, L. Chen, M. T. Ozsü, and D. Zhao, “Dynamic skyline queries in large graphs,” in Proceedings of the Database Systems for Advanced Applications (DASFAA) April 1–4, 2010, Tsukuba, Japan, 2010, pp. 62–78.
- [7] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, “Skyline with Presorting.”
- [8] D. Kossmann, F. Ramsak, and S. Rost, “Shooting stars in the sky: An online algorithm for skyline queries,” in Proceedings of the 28th International Conference on Very Large Data Bases (VLDB) August 20–23, 2001, Hong Kong, China, 2002, pp. 275–286.
- [9] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “Progressive skyline computation in database systems,” *ACM Transactions on Database Systems*, vol. 30, no. 1, 2005, pp. 41–82.
- [10] T. Xia, D. Zhang, and Y. Tao, “On Skylining with Flexible Dominance Relation,” in Proceedings of the 24th International Conference on Data Engineering (ICDE) April 7–12, 2008, Cancun, Mexico, 2008, pp. 1397–1399.
- [11] I.-F. Su, Y.-C. Chung, and C. Lee, “Top-k combinatorial skyline queries,” in Proceedings of the Database Systems for Advanced Applications (DASFAA) April 1–4, 2010, Tsukuba, Japan, 2010, pp. 79–93.
- [12] X. Guo, C. Xiao, and Y. Ishikawa, “Combination Skyline Queries,” *Transactions on Large-Scale Data- and Knowledge-Centered Systems VI*, vol. 7600, 2012, pp. 1–30.
- [13] M. A. Siddique and Y. Morimoto, “Algorithm for computing convex skyline objectsets on numerical databases,” *IEICE TRANSACTIONS on Information and Systems*, vol. E93-D, no. 10, 2010, pp. 0916–8532.
- [14] K. C. K. Lee, B. Zheng, H. Li, and W.-C. Lee, “Approaching the skyline in Z order,” in Proceedings of the 33th International Conference on Very Large Data Bases (VLDB) September 23–27, 2007, Vienna, Austria, 2007, pp. 279–290.
- [15] K.-L. Tan, P.-K. Eng, and B. C. Ooi, “Efficient progressive skyline computation,” in Proceedings of the 27th International Conference on Very Large Data Bases (VLDB) September 11–14, 2001, Rome, Italy, 2001, pp. 301–310.
- [16] C. Y. Chan, H. V. Jagadish, K.-L. Tan, A. K. H. Tung, and Z. Zhang, “Finding k-Dominant Skyline in High Dimensional Space,” in Proceedings of the ACM SIGMOD June 26–29, 2006, Chicago, USA, 2006, pp. 503–514.
- [17] X. Lin, Y. Yuan, W. Wang, and H. Lu, “Stabbing the sky: Efficient Skyline computation over sliding windows,” in Proceedings of the 21th International Conference on Data Engineering (ICDE) April 5–8, 2005, Tokyo, Japan, 2005, pp. 502–513.
- [18] W.-T. Balke, U. Gntzer, and J.-X. Zheng, “Efficient distributed skylining for web information systems,” in Proceedings of the 9th International Conference on Extending Database Technology (EDBT) March 14–18, 2004, Crete, Greece, 2004, pp. 256–273.
- [19] A. Vlachou, C. Doulkeridis, Y. Kotidis, and M. Vazirgiannis, “SKYPEER: Efficient Subspace Skyline Computation over Distributed Data,” in Proceedings of the 23th International Conference on Data Engineering (ICDE) April 15–20, 2007, Istanbul, Turkey, 2007, pp. 416–425.
- [20] Y. Tao, X. Xiao, and J. Pei, “Subsky: Efficient Computation of Skylines in Subspaces,” in Proceedings of the 22th International Conference on Data Engineering (ICDE) April 3–7, 2006, Georgia, USA, 2006, pp. 65–65.
- [21] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “An optimal and progressive algorithm for skyline queries,” in Proceedings of the ACM SIGMOD June 9–12, 2003, California, USA, 2003, pp. 467–478.
- [22] E. Dellis and B. Seeger, “Efficient Computation of Reverse Skyline Queries,” in Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB) September 23–27, 2007, Vienna, Austria, 2007, pp. 291–302.
- [23] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu, “Constructing and exploring composite items,” in Proceedings of the ACM SIGMOD June 6–11, 2010, Indiana, USA, 2010, pp. 843–854.
- [24] Q. Wan, R. C.-W. Wong, and Y. Peng, “Finding top-k profitable products,” in Proceedings of the 27th International Conference on Data Engineering (ICDE) April 11–16, 2011, Hannover, Germany, 2011, pp. 1055–1066.

Cloud Computing System Based on a DHT Structured Using an Hyperbolic Tree

Telesphore Tiendrebeogo

Polytechnic University of Bobo-Dioulasso
Email: tetiendreb@gmail.com

Oumarou Sié

University of Ouagadougou
Email: oumarou.sie@gmail.com

Abstract—During the last decade, Cloud Computing (CC) has been quickly adopted worldwide, and several solutions have emerged. Cloud computing is used to provide storage service, computing power and flexibility to end-users, in order to access data from anywhere at any time. Thus, Cloud Computing is a subscription-based service where you can obtain networked storage space and computer resources. The cloud makes it possible for you to access your information from anywhere at any time. Distributed Hash Table (DHT) plays an important role in distributed systems and applications, particularly in environments distributed on a large scale. In the model of normal Client/Server (C/S model), as we centralize most of the resources on the server, it becomes the most important part as well as the bottleneck and the weak point of the system. On the contrary, distributed model (a typical is Peer-to-Peer (P2P) model) distributes the resources on the nodes in the system. In this paper, we propose a new system of Cloud Computing based on our DHT structured using an hyperbolic tree, which organizes the distributed services so well that peers only need to know part of the system they can get services efficiently. DHT provides two basic operations: retrieves service from DHT and stores service into DHT, which is so simple and graceful, but is suitable for a great variety of resources (Applications, Infrastructures, Platforms), and provides good robustness and high efficiency, especially in large-scale systems. Resources as services are distributed by using virtual coordinates taken in the hyperbolic plane. We use the Poincaré disk model and we perform and evaluate our cloud structure performances. First, we show that our solution is scalable, consistent, reliable. Next, we compare the performances realized by the substitution strategy, which we propose with the strategy of classic replication in a dynamic context.

Keywords—Cloud Computing; DHT; Hyperbolic Tree; Consistent; Reliable; Cloud Services; Storage; Discovery.

I. INTRODUCTION

For decades, extensive work has been done for Distributed Hash Table (DHT) [1] [2] [3] [4]. In academia, researchers have proposed several variants of DHT and improvements, which manage the resources in many kinds of structures, providing abundant choices for the construction of distributed system. Cloud computing (CC) has recently emerged as a compelling paradigm for managing and delivering services over the internet. The rise of Cloud computing is rapidly changing the landscape of information technology, and ultimately turning the long-held promise of utility computing into a reality. The latest emergence of Cloud computing is a significant step towards realizing this utility computing model since it is heavily driven by industry vendors. Thus, Cloud computing centre provides on-line storage and retrieving functionalities, and the services are distributed to tens of thousands of machines in a distributed way.

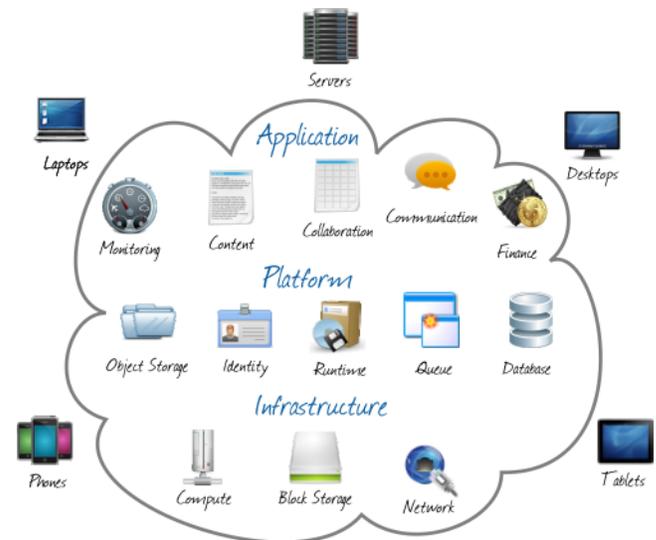


Figure 1. Cloud computing model [5].

In Figure 1, it is important to notice that many devices can interact between them by services exchanges. These services can be as well diverse as varied, going of the monitoring to the network via databases.

Therefore, a distributed storage system seems to be a good choice to manage the gigantic storage. Many big players in the software industry, such as Microsoft, as well as other Internet technology heavyweights, including Google and Amazon, are joining the development of cloud services [6] [7] [8] [9] [10] [11]. Several businesses, also those not technically oriented, want to explore the possibilities and benefits of CC [12]. However, there is a lack of standardization of Cloud computing and Services (CCSs) [7] [8] [13], which makes interoperability when working with multiple services or migrating to new services difficult. Further, there is a big marketing hype around CC, where on-line service providers re-brand their products to be part of the cloud movement [14].

There are a certain number of characteristics associated to the Cloud computing, as follow: variety of resources, Internet centric, virtualization, scalability, automatic adaptation, resource optimization, service SLAs (Service-Level Agreements) and infrastructure SLAs [15]. The Cloud computing corresponds to a virtual computation resource with the possibility to maintain and to manage by itself. From a structural point of view, the resources may be (Figure 2):

- 1) IaaS (Infrastructure as a Service).
- 2) PaaS (Platform as a Service).
- 3) SaaS (Software as a Service).

In Figure 2, the red layers represent layers managed by the providers of cloud resources and the green layers represent those managed by the users of services. Thus, it presents the services characteristics as shown by Chunye et al. [16]. Some defining characteristics of SaaS include:

- 1) Web access to commercial software.
- 2) Software is managed from a central location.
- 3) Software delivered in a “one to many” model.
- 4) Users not required to handle software upgrades and patches.
- 5) Application Programming Interfaces (APIs) allow for integration between different pieces of software.

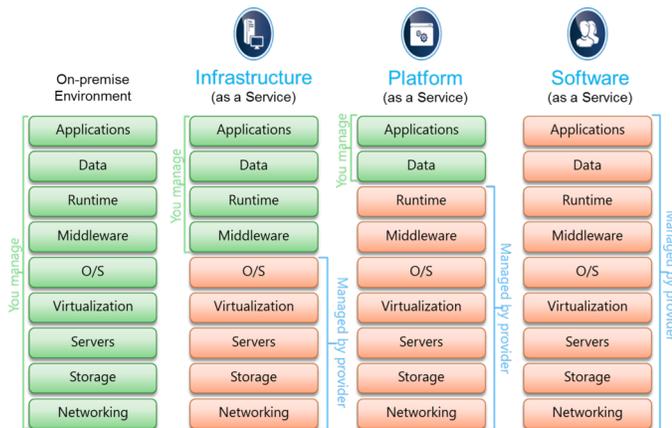


Figure 2. Cloud services model [17].

There are a number of different takes on what constitutes PaaS, but some basic characteristics include:

- 1) Services to develop, test, deploy, host and maintain applications in the same integrated development environment. All the varying services needed to fulfill the application development process.
- 2) Web based user interface creation tools help to create, modify, test and deploy different User Interface (UI) scenarios.
- 3) Multi-tenant architecture where multiple concurrent users utilize the same development application.
- 4) Built in scalability of deployed software, including load balancing and failover.
- 5) Integration with web services and databases via common standards.
- 6) Support for development team collaboration - some PaaS solutions include project planning and communication tools.
- 7) Tools to handle billing and subscription management.

As with the two previous (SaaS and PaaS), IaaS is a rapidly developing field. That said, there are some core characteristics, which describe what IaaS is. IaaS is generally accepted to comply with the following:

- 1) Resources are distributed as a service.
- 2) Allows for dynamic scaling.

- 3) Has a variable cost, utility pricing model.
- 4) Generally includes multiple users on a single piece of hardware.

Thus, for example, it may realize them for a lot of large-scale server cluster structures, including computation servers, storage servers, the bandwidth resources [18]. Cloud platform permits to manage both a large number of computer resources and to store a large number of data. Resource allocation is made in Cloud platform such as user feel that he uses an infinitive amount of resources. In this paper, we make the following contributions:

- 1) We introduce a new structure of Cloud computing using an hyperbolic tree, in which each node is associated with a resource server's and takes of virtual coordinates into hyperbolic space of the model of the Poincaré disk.
- 2) We show how Cloud infrastructures can communicate by greedy routing algorithm using [19].
- 3) We present naming and binding principle in our solution.
- 4) We perform some simulations, furthermore:
 - We show that this structure provides scalability and consistence in database services like data storage and retrieving.
 - We present the results concerning the Floating point precision.
 - We propose a new strategy of resistance in the phenomenon of churn, which we called substitution strategy and we show that it allows to reach satisfactory results in terms of the success rate of the storage and lookup queries of services. Indeed, this strategy allows to improve the availability of the resources on the servers of services.

The remainder of this paper is organized as follows. Section II gives a brief overview of the related previous work. Section III highlights some properties of the hyperbolic plane when represented by the Poincaré disk model. Section IV defines the local addressing and greedy routing algorithms of cloud computing system. Section V defines the binding algorithm of our Cloud system. Section VI presents the results of our practicability evaluation obtained by simulations and we conclude in Section VII.

II. RELATED WORK

The main differences between the CCSs that are deployed are related to the type of service offered, such as storage space and computing power, platforms for own software deployment, or online software applications, ranging from web-email to business analysis tools. Cloud infrastructure services typically offer virtualization platforms, which are an evolution of the virtual private server offerings that are already known for years [20]. The offers of Cloud platform services include the use of the underlying infrastructure, such as servers, network, storage or operating systems, over which the customers have no control, as it is abstracted away below the platform [20] [21]. Cloud software offerings typically provide specific, already-created applications running on a cloud infrastructure, such as simple storage service [22]. A very well known SaaS is the web-based e-mail. Most software CCSs are web-based

applications, which can be accessed from various client devices through a thin client interface, such as a web browser. An Internet-based storage system with strong persistence, high availability, scalability and security is required. Obviously, the centralized methods is not a good way because it lacks of scalability and has the single point of failure problem. If the centre fails, all the owners lose the capability to access their data, which may cause inestimable losses. Besides, it is impossible to store all the data on one machine, though it is facility for management. Even in the cloud computing centre, which provides on-line storage functionality, the data is distributed to tens of thousands of machines in a distributed way. Therefore, a distributed storage system seems to be a good choice to manage the gigantic storage.

How to organize so many kinds of data efficiently is the first hit. DHT with its wonderful structure is suitable to the distributed environment. DHT provides a high efficient data management scheme that each node in the system is responsible to a part of the data. It supports exact and quick routing algorithm to ensure users retrieving their data accurately and timely. Furthermore, replication and backup, fault-tolerant and data recovery, persistent access and update, which are concerned in the storage area are not difficult to DHT. Recently, many researches have proposed a lot of systems such as Chord based Session Management Framework for Software as a Service Cloud (CSMC) [23], MingCloud based on Kademlia algorithm [24], Search on the Clou which is built on Pastry [25]. These systems are based on the DHT structure such as MSPastry [26], Tapestry [27], Kademlia [28], CAN, and Chord [29].

Furthermore, an Efficient Multi-dimensional Index with Node Cube for Cloud computing system [30] has been proposed by Xiangyu Zhang et al., also the RT-CAN index in their Cloud database management system in 2010 [31] has been built by Jinbao Wang et al. Both these two schemes are based respectively on k-d tree and R-tree.

Our work follows the framework proposed in [32]. However, to support multi-dimensional data, new routing algorithms and storage and retrieve queries processing algorithms are proposed. Furthermore, our indexing structure reduces the amount of hops for transfer inside the Cloud and facilitates the deployment of database back-end applications.

III. PRACTICAL USE OF THE HYPERBOLIC GEOMETRY

In this section, we present some properties concerning hyperbolic geometry. Geometries Hyperbolic show similarity with regard to the Euclidian geometry. Both have the same concepts of distances and angles, and they have many common theorems. The two-dimensional hyperbolic plane is the simplest hyperbolic space and has a constant negative curvature equal to -1 as opposed to the Euclidean space, which is not curved. The model that we use to represent the hyperbolic plane is called the Poincaré disk model. In this model, each point is referred by complex coordinates. Beardon, Kleinberg, and Krioukov have detailed all the concepts necessary to understand the hyperbolic plane [33]–[35].

Next, each line of \mathbb{H}^2 splits the plane in several areas as in the Euclidean plane, but there are a certain number of differences. In the Euclidean space, one of elementary properties is the impossibility to create more than two half planes without having them intersect. Our embedding is based

on the geometric property of the hyperbolic plane, which allows to create distinct areas called half planes. As explained by Miquel in [36], in the hyperbolic plane, we can create n half spaces pair wise disjoint whatever n . Our embedded algorithm is based on this property (red line in Figure 3).

Another essential property is that we can split the hyperbolic plane with polygons of any sizes, called p -gons. Each tessellation is represented by a notation of the form $\{p, q\}$ where each polygon has p sides with q of them at each vertex. This form is called a *schläfli symbol*. There exists a hyperbolic tessellation $\{p, q\}$ for every couple $\{p, q\}$ that satisfy the following relation: $(p - 2) * (q - 2) > 4$.

In a splitting, p is the number of sides of the polygons of the *primal* (the black edges and green vertices in Figure 3) and q is the number of sides of the polygons of the *dual* (the red triangles in Figure 3).

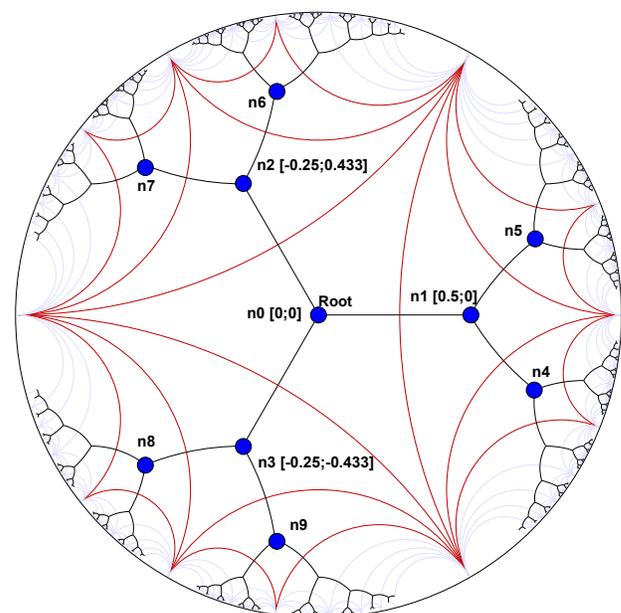


Figure 3. 3-regular tree in the hyperbolic plane.

Our method is to partition the plane and address each node uniquely. We set p to infinity, thus transforming the primal into a regular tree of degree q . The dual is then tessellated with an infinite number of q -gons. This particular tiling splits the hyperbolic plane in distinct spaces and constructs our embedded tree. Figure 3 is an example of a hyperbolic tree with $q = 3$ associated with the Poincaré disk model, in which the hyperbolic plane is represented by an open unit disk of radius 1. Thus, in this model:

- Points are represented by Euclidean points within this unit disk.
- Lines are arcs of circles intersecting the disk and meeting its boundaries at right angles.
- Distances between any two points z and w in the hyperbolic plane ($d_{\mathbb{H}}$) are given by curves minimizing the distance between these two points and are called geodesics of the hyperbolic plane.

To compute hyperbolic distance means to use the Poincaré metric, which is an isometric invariant:

$$d_{\mathbb{H}}(z, w) = \operatorname{argcosh}(1 + 2\Delta) \quad (1)$$

with:

$$\Delta = \frac{|z - w|^2}{(1 - |z|^2)(1 - |w|^2)} \quad (2)$$

because:

$$\sinh^2 \frac{1}{2} d_{\mathbb{H}}(z, w) = \Delta$$

and

$$\cosh^2 \frac{1}{2} d_{\mathbb{H}}(z, w) = \frac{|1 - z\bar{w}|^2}{(1 - |z|^2)(1 - |w|^2)}$$

For more details on the Poincaré metric we refer the reader to the proof in [33]. The hyperbolic distance $d_{\mathbb{H}}(z, w)$ is additive along geodesics and is a Riemann's metric. The authors of [33] do the sketch of an important property for greedy routing: the strict inequality in the triangle inequality. The following relation permits to compute this metric:

$$\tanh \frac{1}{2} d_{\mathbb{H}}(z, w) = \frac{|z - w|}{|1 - z\bar{w}|} \quad (3)$$

where \bar{w} is the complex conjugate of w .

In theoretical perspective, the hyperbolic plane is unlimited. However, to carry out measurements, it is necessary to use a modelled representation of this plane and to define a precision threshold for the calculations.

IV. GREEDY ROUTING IN THE HYPERBOLIC PLANE

In this section, we present how we create the hyperbolic addressing tree for cloud services storage and how we localize these latters in our cloud. We propose here a dynamic, scalable, and reliable hyperbolic greedy routing algorithm [37]. The first step in the creation of our cloud computing based on hyperbolic-tree of services nodes is to start the first services nodes and to define the degree of the addressing tree.

We recall that the hyperbolic coordinates (i.e., a complex number) of a services' servers of the addressing tree are used as the corresponding address to the services' servers in the cloud. A node of services of the tree can provide the addresses corresponding to its children in the hyperbolic-tree. The degree of this latter determines how many addresses each services' servers will be able to attribute for news servers of services connexions. The degree of the hyperbolic-tree is chosen at the beginning for all the lifetime of the cloud. Our cloud architecture is then built incrementally, with each new services' server joining one or more existing servers of services. Over time, the servers of services will leave the overlay until there are no more services' servers in the cloud. So, for every service that must be stored in the system, a Service's Identifier (SID) is associated with it and map then in key-value pair. The key will allow to determine in which servers of services will be stored (like in the Section V). Furthermore, when a service is deleted, the system must be able to update this operation in all the system by forwarding query through the latter. This method is scalable and reliable because unlike [38], we do not have to make a two-pass algorithm over the whole cloud to find its

highest degree. Also, in our system, a server can connect to any other server at any time in order to obtain an address.

The first step is thus to define the degree of the tree because it allows building the *dual*, namely the regular $q - gon$. We nail the root of the tree at the origin of the *primal* and we begin the tiling at the origin of the disk in function of q . Each splitting of the space in order to create disjoint subspaces is ensured once the half spaces are tangent; hence, the *primal* is an infinite q -regular tree. We use the theoretical infinite q -regular tree to construct the greedy embedding of our q -regular tree. So, the regular degree of the tree is the number of sides of the polygon used to build the *dual* (see Figure 3). In other words, the space is allocated for q child servers of services. Each services' server repeats the computation for its own half space. In half space, the space is again allocated for $q - 1$ children. Each child can distribute its addresses in its half space. Algorithm 1 shows how to compute the addresses that can be given to the children of a services' server. The first services' server takes the hyperbolic address (0;0) and is the root of the tree. The root can assign q addresses.

Algorithm 1 Computing the coordinates of a services' servers children.

```

1: procedure CALCCHILDRENCOORDS(server,  $q$ )
2:    $step \leftarrow \operatorname{argcosh}(1/\sin(\pi/q))$ 
3:    $angle \leftarrow 2\pi/q$ 
4:    $childCoords \leftarrow server.Coords$ 
5:   for  $i \leftarrow 1, q$  do
6:      $ChildCoords.rotationLeft(angle)$ 
7:      $ChildCoords.translation(step)$ 
8:      $ChildCoords.rotationRight(\pi)$ 
9:     if  $ChildCoords \neq server.ParentCoords$  then
10:      STORECHILDCOORDS( $ChildCoords$ )
11:    end if
12:  end for
13: end procedure

```

Algorithm 2 Query's greedy routing algorithm in the cloud.

```

1: function GETNEXTHOP(services_S, query) return services_S
2:    $w = query.destServCoords$ 
3:    $m = services\_S.Coords$ 
4:    $d_{min} = \operatorname{argcosh}\left(1 + 2\frac{|m-w|^2}{(1-|m|^2)(1-|w|^2)}\right)$ 
5:    $p_{min} = services\_S$ 
6:   for all  $server\_N \in services\_S.Neighbors$  do
7:      $n = server\_N.Coords$ 
8:      $d = \operatorname{argcosh}\left(1 + 2\frac{|n-w|^2}{(1-|n|^2)(1-|w|^2)}\right)$ 
9:     if  $d < d_{min}$  then
10:       $d_{min} = d$ 
11:       $p_{min} = server\_N$ 
12:    end if
13:  end for
14:  return  $p_{min}$ 
15: end function

```

This distributed algorithm ensures that each services' server is contained in distinct spaces and has unique coordinates.

All the steps of the presented algorithm are suitable for distributed and asynchronous computation. This algorithm

allows the assignment of addresses as coordinates in dynamic topologies. As the global knowledge of the cloud is not necessary, a new services' server can obtain coordinates simply by asking an existing services' server to be its parent and to give it an address for itself. If the asked services' server has already given all its addresses, the new server must ask an address to another existing database server. Besides, when a new services' server obtains an address, it computes the addresses (i.e., hyperbolic coordinates) of its future children. The addressing hyperbolic-tree is thus incrementally built at the same time than the cloud.

When, a new services' server has connected to servers already inside the cloud and has obtained an address from one of those servers, it can start sending requests to store or lookup service in the cloud. The routing process is done on each services' server on the path (starting from the sender) by using the greedy Algorithm 2 based on the hyperbolic distances between the servers. When, a query is received by a services' server, this latter computes the distance from each of its neighbours to the destination and forwards the query to its neighbour, which is the closest to the destination (destination services' servers computing is given in Section V). If no neighbour is closer than the server itself, the query has reached a local minima and is dropped.

In a real network environment, link and services' servers failures can occur often. If the addressing hyperbolic-tree is broken by the failure of a services' server or link, we flush the addresses attributed to the servers beyond the failed server or link and reassign new addresses to those servers (some servers may have first to reconnect to other servers in order to restore connectivity). But, in this paper, we have not detailed this solution.

V. NAMING AND BINDING IN THE HYPERBOLIC PLANE

In this section, we explain how our cloud system stores and retrieves the resources by using these latter's names, which is mapped to its address (virtual coordinates where it is possible to find servers of services). Our solution uses a structured DHT system associated to the virtual addressing mechanism of servers of services and to the greedy routing algorithms presented in Section IV. At startup, each new resource server uses a name that identifies the service (Application, Platform, Infrastructure) that is shared in the system. This name will be kept by the resource server containing the service during all the lifetime of the cloud. When the new resource server obtains an address, it stores the names of these services on different others resources servers. This storage uses the structured DHT of our cloud to store a fragment of key obtain by hashing of service name (explain in the follow). If the same sub-key is already stored in the cloud, an error message is sent back to the resource server containing concerned service in order to change this service name. Thus, the DHT structure used in our cloud itself ensures that services names are unique.

A (name, address) pair, with the name mapping as a key is called a *binding*. Figure 4 shows how and where a given binding is stored in the cloud.

A binder is any resource server that stores these pairs. The depth of a peer in the hyperbolic addressing tree can be defined as the number of parent peers to go through for reaching the root of the tree (including the root itself). When the cloud system is created, a maximum depth for the potential binders

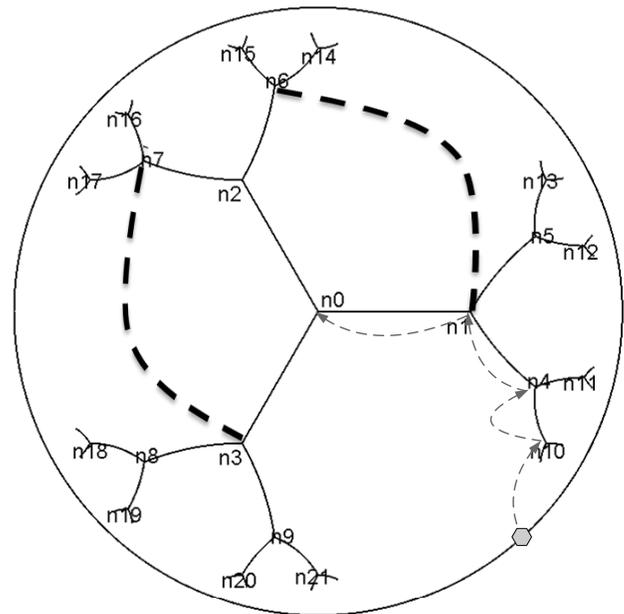


Figure 4. Hyperbolic cloud system.

Algorithm 3 Lookup algorithm in general context for inserting, deleting and updating of service

```

1: function LOOKUPPROCESS(PrimeResourceServer,
   Query) return Service
2:   SID ← Tg.GetQuerySID()
3:   Key ← Hash(QuerySID)
4:   for A do r ∈ RCircular
5:     d ← PMax
6:     i ← 1
7:     while i ≤ [1/2 × log(N)/log(q)] && d ≥ 0 do
8:       TgServAd[r][d] ← GetValue(Key)
9:       Service ← GetValue(TgServAd[r][d], SID)
10:      if Service ≠ null then
11:        if Query == delete then
12:          delete(SID)
13:        end if
14:        if (Query == update) then
15:          update(SID)
16:        end if
17:        if Query == select then
18:          return Service
19:          break
20:        end if
21:        i ++
22:      end if
23:      d --
24:    end while
25:  end for
26:  return 0
27: end function

```

is chosen. This depth permits to define the maximum number of servers of services that can connect and share different services. Also, the depth *d* is chosen such it minimizes the

Inequality (4):

$$p \times \left(\frac{(1 - (p - 1)^d)}{2 - p} + 1 \right) \geq N \tag{4}$$

with p the degree of our hyperbolic tree. Thus, this value is defined as the *binding tree depth*.

When a new resource server joins the cloud by connecting to other resources servers, it obtains an address from one of these resources servers. So, each service name of the resource server is mapped into key by hashing its name with the SHA-2 algorithm (SHA-2 gives 512-bit key). Next, the new resource server divides the 512-bit key into 16 equally sized 32-bit sub-keys (for redundancy storage). The resource server selects the first sub-key and maps it to an angle by a linear transformation. The angle is given by:

$$\alpha = 2\pi \times \frac{32\text{-bit sub-key}}{0xFFFFFFFF} \tag{5}$$

The resource server then computes a virtual point v on the unit circle by using this angle:

$$v(x, y) \text{ with } \begin{cases} x = \cos(\alpha) \\ y = \sin(\alpha) \end{cases} \tag{6}$$

Next, the resource server determines the coordinates of the closest binder to the computed virtual point above by using the given *binding tree depth*. In Figure 4, we set the *binding tree depth* to three to avoid cluttering the figure. It is important to note that this closest service (name of the binder) may not really exist. If no resource server is currently owning this address, this latter, then sends a stored query (containing service name and address) to this closest resource server. This query is routed inside the cloud by using the greedy algorithm of Section IV. If the query fails because the binder does not exist or because of node/link failures, it is redirected to the next closest binder, which is the father of the computed binder. This process continues until the query reaches to the root resource server having the address (0;0) (which is the farthest binder) or the number of resources servers is given by the following relation (radial strategy):

$$S \leq \left\lfloor \frac{1}{2} \times \frac{\log(N)}{\log(q)} \right\rfloor \tag{7}$$

with N equal to number of servers of services, q to degree of hyperbolic-tree.

This strategy permits to not saturate the closest servers of root's server of the hyperbolic tree. Indeed, when the replication of service go down towards the hyperbolic tree root's, the different paths from edge of unit circle towards root tighten and provoke a saturation of the servers of services. Inequality (7) permits to reduce this risk, and gives a better load balancing of services storage in our cloud. Thus, we search with this method, to increase the number of copies across a path of the hyperbolic tree. So, this strategy has a double interest:

- Allows to make the services available.
- Limits the saturation of the servers as a result of large number of services query and the same reduces the time of response to the servers.

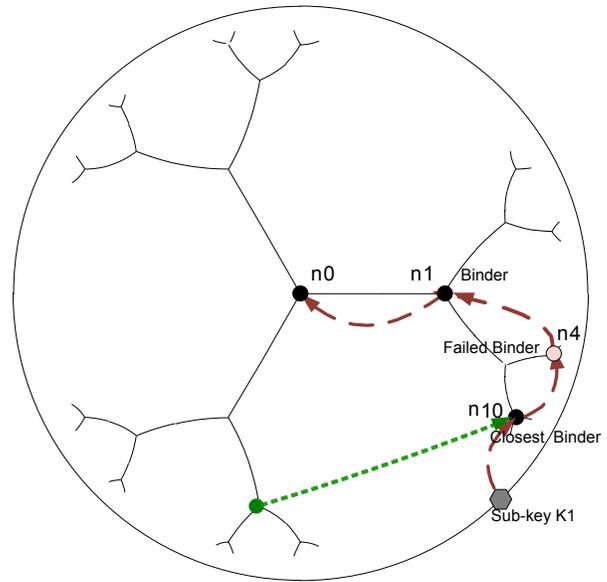


Figure 5. Radial replication in the cloud.

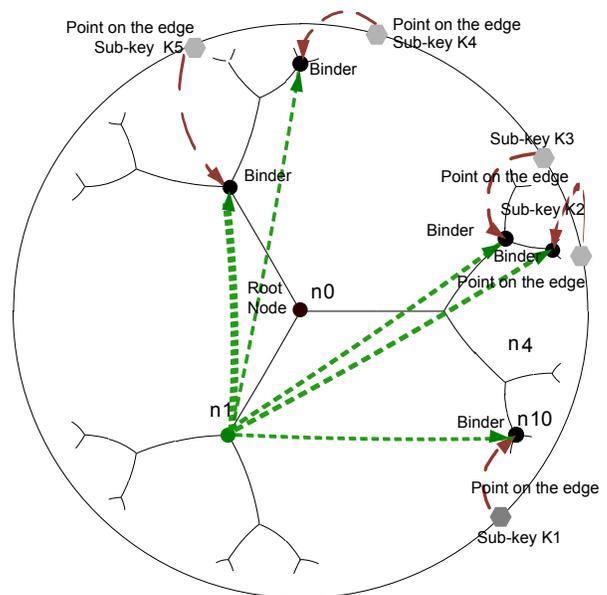


Figure 6. Circular replication in the cloud.

To reduce the impact of the dynamic (the leave or the join of the services server's also the adding or the deleting of the service) of the cloud, uses a redundancy mechanism that consists to increase the number of copies. The number of stored copies of a pair along the binding radius may be an arbitrary value (chosen to the cloud creating) set at the overlay creation. Similarly, the division of the key in 16 sub-keys is arbitrary and could be increased or reduced depending on the redundancy (circular strategy) needed.

To conclude, we can define two redundancy mechanisms (represented by Figure 5 and Figure 6) for storing copies of a given binding:

- 1) radial strategy of replication.
- 2) circular strategy of replication.

Algorithm 4 Services storage's processing in the general context

```

1: function STOREPROCESS(PrimeServServer Query)
   return 0
2:   SID ← Query.GetSID()
3:   Key ← Hash(SID)
4:   for A doll red ∈ RCircular
5:     depth ← PMax
6:     i ← 1
7:     while  $i \leq \left\lfloor \frac{1}{2} \times \frac{\log(N)}{\log(q)} \right\rfloor$  && depth ≥ 0 do
8:       d ← depth
9:       r ← red
10:      S_Key[r][d] ← CompS_key(Key)[r][d]
11:      TgServAd[r][d] ← CompAd(S_Key[r][d])
12:      TgServ ← GetTg(TgServAd[r][d])
13:      if route(Query, TgServ) then
14:        i ++
15:        put(SID, Query)
16:      end if
17:      d --
18:    end while
19:  end for
20:  return 0
21: end function

```

These mechanisms enable our cloud system to cope with a non-uniform growth of the system and they ensure that a service will be stored in a redundant way that will maximize the success rate of its retrieval. Our solution has the property of consistent hashing: if one resource server (or a service of the resource server) fails, only its keys are lost, but the other binders are not impacted and the whole cloud system remains coherent. As in many existing systems, pairs will be stored by following a hybrid soft and the hard state strategy.

A pair (service name, address) will have to be stored by its creator every x period of time, otherwise it will be flushed by the binders that store it. A delete message may be sent by the creator to remove the pair before the end of the period. Algorithm 4 describes services storage's process on the servers y using Service IDentifier.

When a user wants to use a service, it is connected to any resource server on the cloud and chosen the service name's. Next, current service server's, by using previous mechanism hashes name and finds associated service by sending retrieving query in the cloud. Retrieving query is processed and different servers where is stored the service are located. Then, it is possible to execute one of following operation (Algorithm 3):

- Deleting: deleting operation allows to make unavailable all the occurrences of a service through replication in the system,
- Updating: it allows to bring changes in a service through all these replications,
- Inserting: this allows to add a new service to the the cloud system (i.e., in all resources servers of replication).

Altogether, the mechanisms of storage and lookup that we propose are (on one hand) reliable, in the sense that we always

manage to store or to find any service in our system of cloud. On the other hand, this system is strong because it assures an availability of the services in a strongly dynamic environment.

VI. SIMULATIONS

In this section, we present the results of the simulations that we have performed and we have assessed the practicability, and in some cases the scalability, of our addressing. Our cloud system is considered as dynamic (the phenomenon of churn is applied). We use the Peersim [39] simulator for cloud computing system simulation and it allows to service name following the uniform distribution. Our simulation involves the following parameters of the DHT used in our cloud (Excepted, the study on the floating point precision issue). These parameters are valid for all the DHTs that we compare:

- Number of servers of services connected and used to store different services is equal to 10000 in the starting up, maximum number of services by server equal to 2000.
- We consider that our cloud system is dynamic and the rate of churn varying from 10% to 60%.
- We consider a simulation performed during 2 hours.
- The leaving and joining of the servers of services follow an exponential distribution as well as the inserting and the deleting of the services of the clouds associated.
- We suppose that the system receives 6 millions of queries following an exponential distribution with a median equal to 10 minutes.

A. Characteristics of our hyperbolic tree

1) *Analysis of the angular gap in the Poincaré disk*: In this part, we try to analyse the average gap between points computed on the unity's circle.

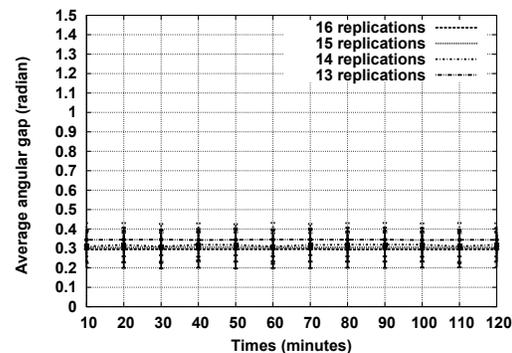


Figure 7. Angular gap from 13 to 16 circular replications.

This study will allow us to see the level of efficiency of our strategy of circular replication. Indeed, according to the number of chosen circular replications, in our case 16, there is a threshold for which the probability two primary binders is confused.

Except, if two binders at least are confused, it means reducing of number of replication actually made by the system

and of the same cost to reduce the rate of the storage and lookup queries.

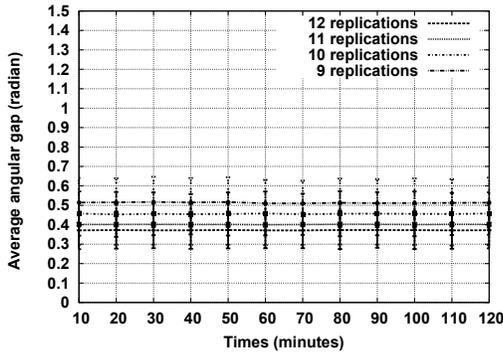


Figure 8. Angular gap from 9 to 12 circular replications.

Figures 7-10 show in the context of our simulation the average gaps between points calculated of the circle unity as well as their standard deviations.

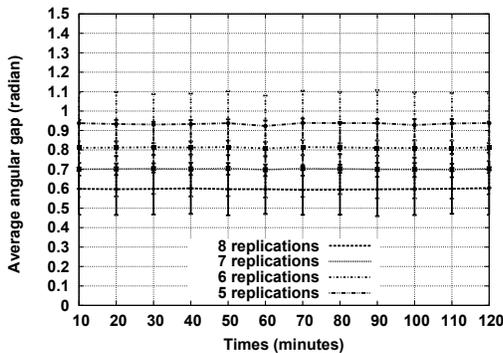


Figure 9. Angular gap from 5 to 8 circular replications.

In a general way for all the figures, we notice a growth of the average value of the angular gaps, which is inversely proportional at the level of used replication. Indeed, we have a variation of 0.29 for 15 replications in approximately 1.4 out of 2 replications.

This indicates that our system behaves well because even for a large number of circular replications, the angular gap is enough important so that we obtain (in most of the time) different primary binders. Furthermore, for example, we can notice that in Figure 7, which concerns the average angular gaps in levels of replication between 16 and 13, the values remain constant during all the period of simulation.

So, we have gaps between 0.29 and 0.35 with standard deviations about 30% of the average. This indicates that approximately 78% of the calculated points have an angular gap between 0.20 and 0.40 for 16 replications and between 0.24 and 0.46 for 13 replications. So, in Figures 7-10 the averages seem to be significant because, in each case, we

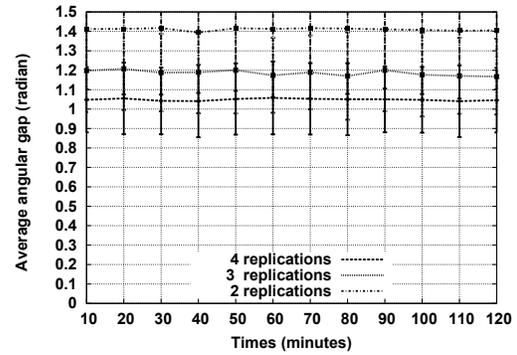


Figure 10. Angular gap from 2 to 4 circular replications.

observe a standard deviation at the most equal to 30% of average.

These results tend to confirm that our strategy of circular replication is reliable in the measure or at a given level of replication, we can associate various primary binders, so avoiding that a departure of stacker not in the loss of information of a large number of nodes.

2) *Analysis of the number of sub-keys*: Figure 11 presents the evolution of the number of primary binders according to the number of sub-keys chooses for the simulation.

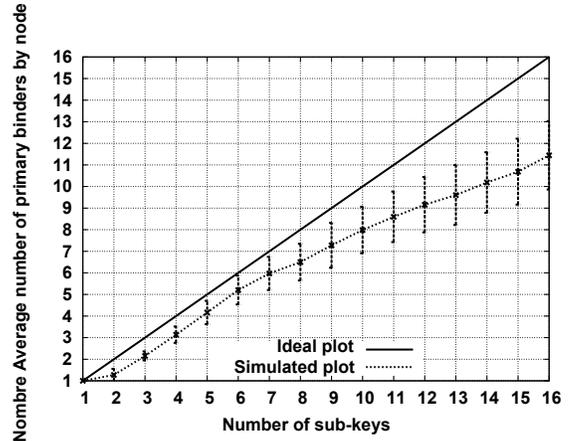


Figure 11. Variation of binder number depending of the number of sub-keys

This curve shows a continuous growth of the number of binders according to the number of money keys. Furthermore, we can observe that this curve is below the ideal case or the number of binders always corresponds among key money. The feigned curve is very close to the ideal curve, which shows that we have a satisfactory situation for all the levels of replication.

B. Load balancing in the cloud

Figure 12 shows an experimental distribution of points corresponding to the scatter plot of the distribution of servers

of services in our cloud. Thus, we can mark that hyperbolic-tree of our cloud system is balanced. Indeed, we can noticed by part and others around the unit circle, which we have servers of services.

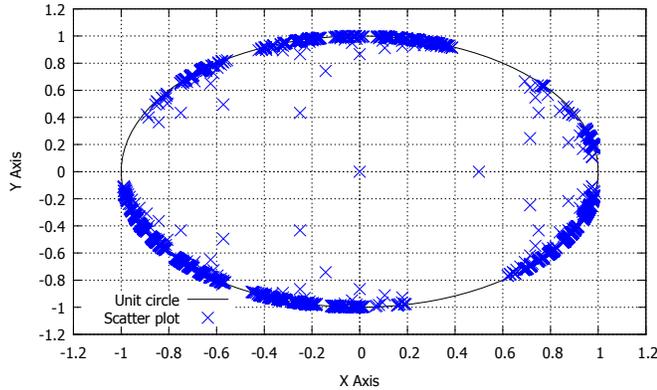


Figure 12. Scatter plot corresponding to the distributed database servers.

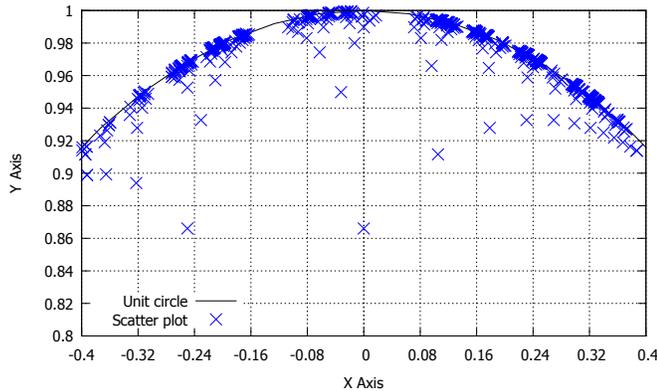


Figure 13. Distribution of nodes in the neighbourhood of the edge of the unit circle.

This has an almost uniform distribution around the root, which implies that our system builds a well-balanced tree what will more easily allow to reach a load balancing of storage.

Figure 13 shows correspondingly Poincaré disk model that no address of resource server belongs on the edge of the unit circle. Indeed, the addresses of resource server were obtained by projection of the tree of the hyperbolic plane in a circle of the Euclidean plane of radius 1 and of centre with coordinates (0;0).

C. Floating point precision issue

One property of the Poincaré model is misleading: the distances are not preserved. If we observe the Poincaré model from an outside point of view, the distance are smaller than the reality (i.e., inside the plane). Because of the model is a representation of the hyperbolic plane in Euclidean plane. Indeed, closer to the boundary of the circle are the points, the farther they are in reality. This phenomenon is illustrated in Figure 13 obtained by simulation.

The hyperbolic plane has a boundary circle at infinity represented in the Poincaré unit disk model (i.e., the open

unit disk) by a circle of radius 1 and centred on the origin O (Section III). The open unit disk around O is the set of points whose the complex modulus is less than 1: $|w| < 1$ with $|w| = \sqrt{(w_{Re})^2 + (w_{Im})^2}$.

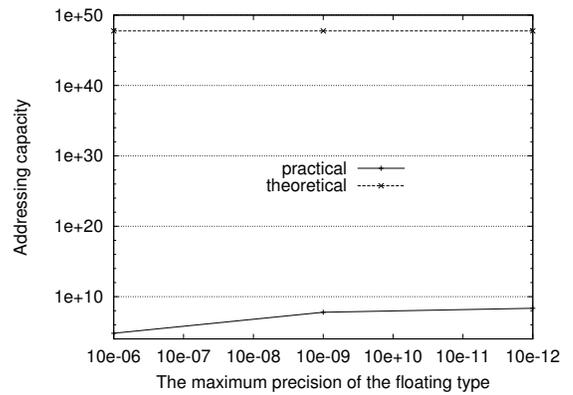


Figure 14. Addressing capacity as a function of the floating point precision threshold.

In practice, an embedding of such a mathematical space is constrained by the precision of the floating type used, typically a *double*. It is a problem of arithmetic precision, we reach the maximal accuracy allowed by the computing in floating point. Indeed, the calculations obey with the IEEE 754 standard, which determines the binary floating point representation. The floating point arithmetic can be implemented with variable length significant that are sized depending on the needs. This is called Arbitrary Precision Arithmetic (APA). To compute with extended precision we have found three computational solutions: computation with rounding (e.g., IEEE 754), interval arithmetic and the Real Ram model.

But we should use a specific library such as the *MultiPrecision Complex MPC* library. As the complexity of using APA is important and as we have enough addressing capacity by using standard floating point numbers, we keep on using the *double* type representation. Thus, two points cannot be closer that the minimum non zero *double*. Hence, the minimal half space is the space that can contain one distinct point.

To address this issue we should answer at the question of Paul Zimmermann: Can we compute on the computer? In order to assess the impact of overlay parameters such as the degree and the depth, we carry out significant simulations, we try to find the best tree parameters to maximize the addressing capacity. This brings us to some practical concerns:

- How to determine the maximum number of subspaces that we can create to assign a coordinate to a node?
- What is the maximal node density in a subspace?

To carry out our practical analysis, we proceed as follows. We embed a tree with a degree of 32 and a depth equal to 32. Then, we assign an address to each node. We show in Figure 14 the gap between the number of addresses in theory and in practice. We set the maximum precision to a given value and compute the addresses. We then vary the maximum precision with the significant digits evolving from 6 to 12 (i.e., 10e-6 to 10e-12). With these characteristics, the theoretical addressing capacity is the same whatever the precision, namely

maximum (as expected). The addressing capacity increases strongly between an accuracy of 6 to 9 digits compared to the transition of 9 to 12 digits because more disjoint points appear.

1) **Influence of the degree of hyperbolic tree:** Finally, for a degree different of 32, the addressing capacity stays closer to 2.246E+08.

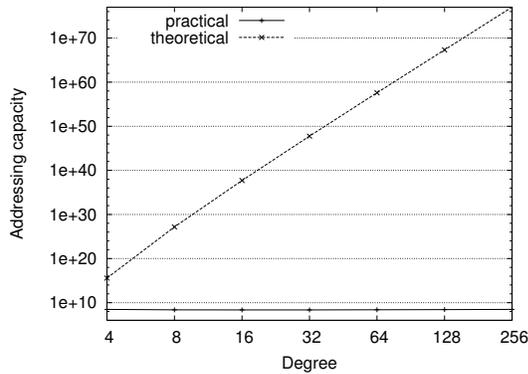


Figure 15. Influence of the degree on the number of theoretical addresses.

To analyse the influence of the degree on the addressing capacity, we use a precision of 12 digits and a tree depth of 32 hops. The tree degree evolves from 4 to 256. Figure 15 shows that the theoretical addressing capacity increases linearly in function of the degree, unlike the “practical” plot that seems constant. In fact, we show in Figure 16 that with a degree higher than 32 the gain is weak compared to the order of magnitude observed in Figure 15.

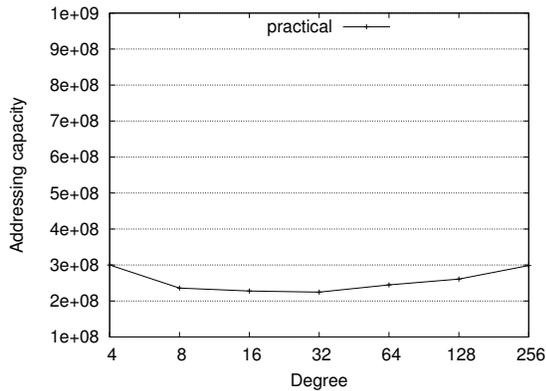


Figure 16. Influence of the degree on the number of practical addresses.

2) **Influence of the depth:** In the same way, we analyse the influence of the depth on the addressing capacity. The precision is the same as the previous one and the tree degree is set to 32. The tree depth evolves from 4 to 32. In Figure 17, the increase of the theoretical addressing capacity is exponential when the depth increases. As expected, this matches with the normal characteristics of \mathbb{H}^2 . On the other side, in practice, the addressing capacity achieves the threshold at 2.246E+08.

The threshold is reached with only a depth of 8. Indeed, the boundary of the disk is quickly reached. We recall that the

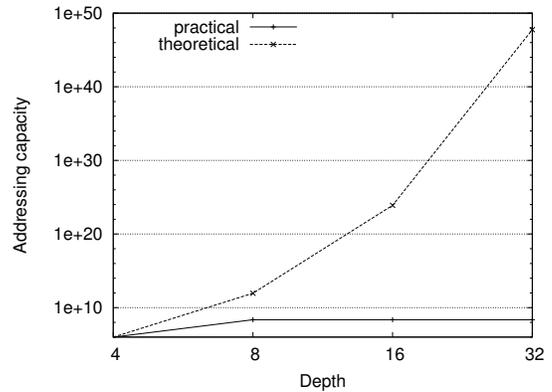


Figure 17. Influence of the depth on the number of addresses.

tiling is built with regular q -gon, q is the number of sides of the q -gon. In \mathbb{H}^2 , whatever q , there exists a distance d from which the created subspaces are pair wise disjoint (i.e., the sides of the q -gon are tangent as the red line in Figure 3). This property is more explained in Section IV. Because of this distance and the precision of the floating type (12 digits), the leaves of the tree can reach the boundary of the disk after only 7 hops (i.e., a depth of 7).

A fine tuning of the degree parameter can improve the addressing capacity, namely we can set q to the degree of the tree that we find the most suitable. This is possible because we create an overlay network, we can have some freedom in setting the overlay links and thus we can restrain the degree of the addressing tree.

D. Performances analysis

1) **Evaluation of the substitution strategy:** Figure 18 shows us the impact of the substitution strategy on the average number of binders reached by every node during the process of storage. Indeed, we compare two situations.

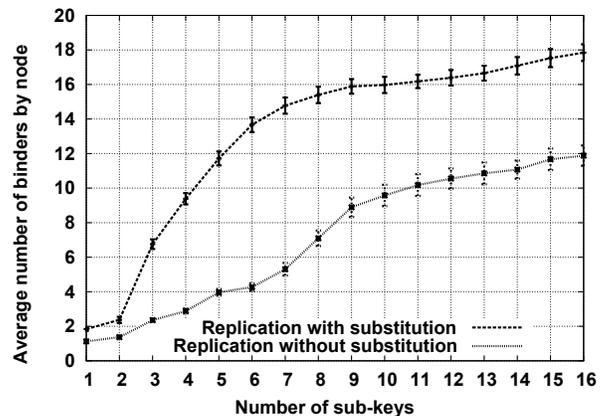


Figure 18. Evaluation of the performances of the substitution strategy

At first, the case where the new resources servers join in a random way the system via the first resource server having free address. Then, the case or the new resources servers try to substitute themselves for binders having left the network to continue to assume this role.

In Figure 18, we notice that according to the number of sub-keys used, average numbers of binders reached in the case of the substitution are much greater than in the case of a connection of new nodes to the first random address. This established fact shows that the substitution strategy returns us a stronger system in the sense that the couples keys-values of the various nodes are distributed on a large number of binders so returning the available information even in case of departure of binders. So, for example, by using 8 sub-keys, we have approximately 7 binders on average during all the simulation in the strategy of circular replication and simple radial road against about 14 binders when we use the substitution method besides the classic replication. Furthermore, for 16 sub-keys, we have approximately 12 binders against about 18 on average in the case of the substitution.

2) **Success rate with the substitution strategy:** Figure 19 presents two plots illustrating respectively the evolution of the rate of success of the storage queries in the case of a classic replication (circular and radial) then in the case of a classic replication with which we associated the substitution strategy.

We can easily note that the level of classic replication, the success rate offered by the substitution strategy is better. Indeed, in a replication, we note a rate of average success about 62% of successes for the classic replication against about 75% of the method of substitution.

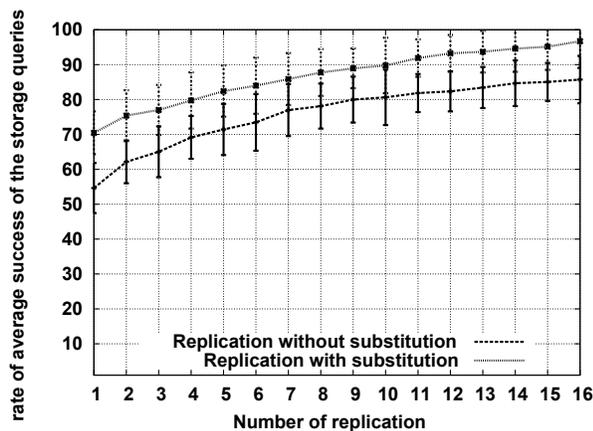


Figure 19. Evaluation of the success rate in storage context

In 7 replications, we note 78% of successes in the case without substitution against 89% in the case using this strategy.

With 15 replications, we observe 85% of success rates in without substitution against 97% of successes in the case of the substitution. Generally, we can notice on average an earnings of about 10% in the success of the storage queries.

These results show the interest in using the strategy of replication, which constitutes an additional factor in the im-

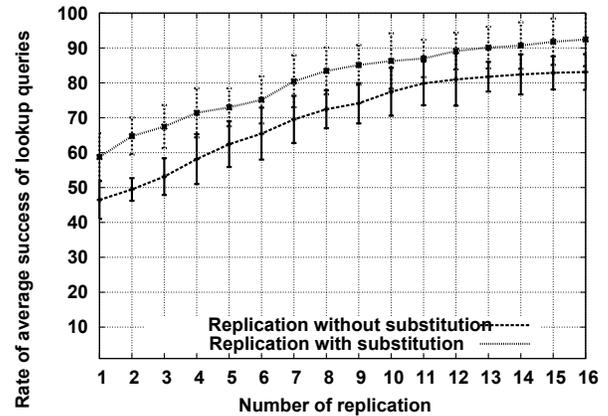


Figure 20. Evaluation of the success rate in lookup context

provement of the success of the storage queries of services (Figure 19). The same strategy impacts positively services lookup, as shown in Figure 20.

In the case of queries for a lookup, the situation is almost similar because, as we can observe in Figure 20, a better rate success when we use the substitution strategy. Indeed, also in it an earnings of an average 9% of successes on queries for a lookup. Whether it is for the storage or for the lookup, we can say that our strategy is reliable because it allows us to increase considerably the success rates.

VII. CONCLUSION

In this paper, we have proposed a new Cloud platform, providing scalable, reliable and robust distribution of services. Indeed, these properties are an essential requirement for this kind of system. There were few research reports proposed for multi-dimensional indexing schemes for Cloud platform to manage the huge and variety services to address the complex queries efficiently (because most of these proposed systems do not have properties given behind). In this study, we have showed that our solution using hyperbolic-tree with virtual coordinates is consistent, reliable, scalable, and hardy, because it supports enough well churn phenomenon. Furthermore, through the study of floating point precision issue, we showed the limits of the capacity of addressing of our hyperbolic tree depending to the degree and to the depth. Thus, our theoretical analysis of the Poincaré disk model has shown that even though we lose a huge number of potential addresses because of the floating point accuracy limits, we can still handle a vast amount of servers of services (i.e., 100M order of magnitude) before reaching those limits of the model. Our system has properties, which allow to guarantee the availability of the Cloud services. Furthermore, it gives very good results when we apply the substitution strategy in churn context. In the future work, we would like to investigate the impact of churn phenomenon issue and thus to develop a new multi-dimensional index structure, which ensures a best result in dynamic context in the aim to supply a variety of cloud services.

REFERENCES

- [1] T. Tiendrebeogo, "A new spatial database management system using an hyperbolic tree," in *DBKDA 2015 : The Seventh International Conference on Advances in Databases, Knowledge, and Data Applications*, May 2015, pp. 43–50.
- [2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications." New York, NY, USA: ACM, 2001, pp. 149–160.
- [3] M. Castro, M. Costa, and A. I. T. Rowstron, "Performance and dependability of structured peer-to-peer overlays," in *International Conference on Dependable Systems and Networks (DSN)*, 28 June - 1 July 2004, Florence, Italy, Proceedings, 2004, pp. 9–18.
- [4] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric." London, UK, UK: Springer-Verlag, 2002, pp. 53–65.
- [5] "Cloud computing model," August, 2015, <https://upload.wikimedia.org/wikipedia/commons/thumb/9/93/Nuage33.png/400px-Nuage33.png>.
- [6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., Feb 2009.
- [7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, Apr. 2010.
- [8] C. Baun, M. Kunze, J. Nimis, and S. Tai, *Cloud Computing: Web-Based Dynamic IT Services*, 1st ed. Springer Publishing Company, Incorporated, 2011.
- [9] P. Murray, "Enterprise grade cloud computing," in *Proceedings of the Third Workshop on Dependable Distributed Data Management*, ser. WDDM '09. New York, NY, USA: ACM, 2009, pp. 1–1.
- [10] A. Weiss, "Computing in the clouds," *netWorker*, vol. 11, no. 4, Dec. 2007.
- [11] D. Hilley, "Cloud computing: A taxonomy of platform and infrastructure-level offerings," Georgia Institute of Technology, Tech. Rep., 2009.
- [12] T. Jowitt, "Four out of five enterprises giving cloud a try." *Computerworld UK* (visited: 2010, May 7), 2009. [Online]. Available: <http://www.computerworlduk.com/management/it-business/services-sourcing/news/index.cfm?newsId=16355>
- [13] R. L. Grossman, "The case for cloud computing," *IT Professional*, vol. 11, no. 2, Mar. 2009.
- [14] A. Plesser, "Four out of five enterprises giving cloud a try," *Tech. Rep.*, 2008. [Online]. Available: <http://www.computerworlduk.com/management/it-business/services-sourcing/news/index.cfm?newsId=16355>
- [15] B. Huang and Y. Peng, "An efficient two-level bitmap index for cloud data management," in *Proceedings of the 3rd IEEE International Conference on Communication Software and Networks (ICCSN)*. IEEE, 2011, pp. 509–513.
- [16] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, "The characteristics of cloud computing," in *Parallel Processing Workshops (ICPPW)*, 2010 39th International Conference on, Sept 2010, pp. 275–279.
- [17] "Cloud services model," August, 2015, <https://www.simple-talk.com/cloud/development/a-comprehensive-introduction-to-cloud-computing/>.
- [18] S. Zhang, S. Zhang, X. Chen, and S. Wu, "Analysis and research of cloud computing system instance," in *Proceedings of the 2010 Second International Conference on Future Networks*, ser. ICFN '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 88–92. [Online]. Available: <http://dx.doi.org/10.1109/ICFN.2010.60>
- [19] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 96–108. [Online]. Available: <http://doi.acm.org/10.1145/938985.938996>
- [20] D. Hilley, "Cloud computing: A taxonomy of platform and infrastructure-level offerings," *Tech. Rep.*, 2009.
- [21] P. M. Mell and T. Grance, "Sp 800-145. the nist definition of cloud computing," Gaithersburg, MD, United States, Tech. Rep., 2011.
- [22] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon s3 for science grids: A viable solution?" in *Proceedings of the 2008 International Workshop on Data-aware Distributed Computing*, ser. DADC '08. New York, NY, USA: ACM, 2008, pp. 55–64. [Online]. Available: <http://doi.acm.org/10.1145/1383519.1383526>
- [23] Z. Pervez, A. M. Khattak, S. Lee, and Y.-K. Lee, "Csmc: Chord based session management framework for software as a service cloud," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, ser. ICUIMC '11. New York, NY, USA: ACM, 2011, pp. 30:1–30:8. [Online]. Available: <http://doi.acm.org/10.1145/1968613.1968650>
- [24] J.-Y. Wu, J.-I. Zhang, T. Wang, and Q.-I. Shen, "Study on redundant strategies in peer to peer cloud storage systems." *Applied Mathematics Information Sciences*, 2011, pp. 235–242.
- [25] D. S. and K. Piromsopa, "Research on cloud storage environment file system performance optimization," vol. 2. *Information Management, Innovation Management and Industrial Engineering (ICIII)*, 2011, pp. 58–62.
- [26] C. Miguel, C. Manuel, and R. Antony, "Performance and dependability of structured peer-to-peer overlays." in *proceedings of the 2004 DSN (IEEE)*, 2004, pp. 9–18.
- [27] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," vol. 22, 2004, pp. 41–53.
- [28] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK, UK: Springer-Verlag, 2002, pp. 53–65. [Online]. Available: <http://dl.acm.org/citation.cfm?id=646334.687801>
- [29] N. Antonopoulos, J. Salter, and R. M. A. Peel, "A multi-ring method for efficient multi-dimensional data lookup in p2p networks." in *FCS, H. R. Arabnia and R. Joshua, Eds. CSREA Press*, 2005, pp. 10–16. [Online]. Available: <http://dblp.uni-trier.de/db/conf/fcs/fcs2005.html#AntonopoulosSP05>
- [30] X. Zhang, J. Ai, Z. Wang, J. Lu, and X. Meng, "An efficient multi-dimensional index for cloud data management," in *Proceedings of the First International Workshop on Cloud Data Management*, ser. CloudDB '09. New York, NY, USA: ACM, 2009, pp. 17–24. [Online]. Available: <http://doi.acm.org/10.1145/1651263.1651267>
- [31] J. Wang, S. Wu, H. Gao, J. Li, and B. C. Ooi, "Indexing multi-dimensional data in a cloud system," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 591–602. [Online]. Available: <http://doi.acm.org/10.1145/1807167.1807232>
- [32] S. Wu and K.-L. Wu, "An indexing framework for efficient retrieval on the cloud." vol. 32, no. 1, 2009, pp. 75–82. [Online]. Available: <http://dblp.uni-trier.de/db/journals/debu/debu32.html#WuW09>
- [33] A. F. Beardon and D. Minda, "The hyperbolic metric and geometric function theory," in *International Workshop on Quasiconformal Mappings And Their Applications*, 2006, pp. 9–56.
- [34] R. Kleinberg, "Geographic routing using hyperbolic space," in *Proceedings of the 26th Annual Joint Conference of INFOCOM. IEEE Computer and Communications Societies*, 2007, pp. 1902–1909.
- [35] D. Krioukov, F. Papadopoulos, M. Boguñá, and A. Vahdat, "Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, 2009, pp. 15–17.
- [36] A. Miquel, "Un afficheur générique d'arbres à l'aide de la géométrie hyperbolique," in *Journées francophones des langages applicatifs (JFLA)*, 2000, pp. 49–62.
- [37] F. Papadopoulos, D. Krioukov, M. Boguñá, and A. Vahdat, "Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces," in *Proceedings of the 29th Conference on Information Communications*, ser. INFOCOM '10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 2973–2981. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1833515.1833893>
- [38] V. Gaede and O. Günther, "Multidimensional access methods," *ACM Comput. Surv.*, vol. 30, no. 2, Jun. 1998, pp. 170–231. [Online]. Available: <http://doi.acm.org/10.1145/280277.280279>
- [39] I. Kazmi and S. F. Y. Bukhari, "Peersim: An efficient & scalable testbed for heterogeneous cluster-based p2p network protocols." Washington,

DC, USA: IEEE Computer Society, 2011, pp. 420–425. [Online].
Available: <http://dx.doi.org/10.1109/UKSIM.2011.86>



www.iariajournals.org

International Journal On Advances in Intelligent Systems

🔗 issn: 1942-2679

International Journal On Advances in Internet Technology

🔗 issn: 1942-2652

International Journal On Advances in Life Sciences

🔗 issn: 1942-2660

International Journal On Advances in Networks and Services

🔗 issn: 1942-2644

International Journal On Advances in Security

🔗 issn: 1942-2636

International Journal On Advances in Software

🔗 issn: 1942-2628

International Journal On Advances in Systems and Measurements

🔗 issn: 1942-261x

International Journal On Advances in Telecommunications

🔗 issn: 1942-2601