

International Journal on Advances in Internet Technology



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

ISSN: 1942-2652

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 Internet Technology, issn 1942-2652
vol. 5, no. 3 & 4, year 2012, http://www.ariajournals.org/internet_technology/

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 Internet Technology, issn 1942-2652
vol. 5, no. 3 & 4, year 2012, <start page>:<end page>, http://www.ariajournals.org/internet_technology/

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 © 2012 IARIA

Editor-in-Chief

Alessandro Bogliolo, Universita di Urbino, Italy

Editorial Advisory Board

Lasse Berntzen, Vestfold University College - Tonsberg, Norway

Michel Diaz, LAAS, France

Evangelos Kranakis, Carleton University, Canada

Bertrand Mathieu, Orange-ftgroup, France

Editorial Board

Jemal Abawajy, Deakin University, Australia

Chang-Jun Ahn, School of Engineering, Chiba University, Japan

Sultan Aljahdali, Taif University, Saudi Arabia

Shadi Aljawarneh, Isra University, Jordan

Giner Alor Hernández, Instituto Tecnológico de Orizaba, Mexico

Onur Alparslan, Osaka University, Japan

Feda Alshahwan, The University of Surrey, UK

Ioannis Anagnostopoulos, University of Central Greece - Lamia, Greece

M.Ali Aydin, Istanbul University, Turkey

Gilbert Babin, HEC Montréal, Canada

Faouzi Bader, CTTC, Spain

Kambiz Badie, Research Institute for ICT & University of Tehran, Iran

Jasmina Baraković Husić, BH Telecom, Bosnia and Herzegovina

Ataul Bari, University of Western Ontario, Canada

Javier Barria, Imperial College London, UK

Shlomo Berkovsky, NICTA, Australia

Lasse Berntzen, Vestfold University College - Tønsberg, Norway

Nik Bessis, University of Derby, UK

Jun Bi, Tsinghua University, China

Marco Block-Berlitz, Freie Universität Berlin, Germany

Christophe Bobda, University of Arkansas, USA

Alessandro Bogliolo, DiSBef-STI University of Urbino, Italy

Thomas Michael Bohnert, Zurich University of Applied Sciences, Switzerland

Eugen Borcoci, University "Politehnica" of Bucharest, Romania

Luis Borges Gouveia, University Fernando Pessoa, Portugal

Fernando Boronat Seguí, Universidad Politécnica de Valencia, Spain

Mahmoud Boufaida, Mentouri University - Constantine, Algeria

Christos Bouras, University of Patras, Greece

Agnieszka Brachman, Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Thierry Brouard, Université François Rabelais de Tours, France
Dumitru Dan Burdescu, University of Craiova, Romania
Carlos T. Calafate, Universitat Politècnica de València, Spain
Christian Callegari, University of Pisa, Italy
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Miriam A. M. Capretz, The University of Western Ontario, Canada
Ajay Chakravarthy, University of Southampton IT Innovation Centre, UK
Chin-Chen Chang, Feng Chia University, Taiwan
Ruay-Shiung Chang, National Dong Hwa University, Taiwan
Tzung-Shi Chen, National University of Tainan, Taiwan
Xi Chen, University of Washington, USA
Dickson Chiu, Dickson Computer Systems, Hong Kong
IlKwon Cho, National Information Society Agency, South Korea
Andrzej Chydzinski, Silesian University of Technology, Poland
Noël Crespi, Telecom SudParis, France
Antonio Cuadra-Sanchez, Indra, Spain
Javier Cubo, University of Malaga, Spain
Alfredo Cuzzocrea, University of Calabria, Italy
Jan de Meer, smartspace®lab.eu GmbH, Germany
Sagarmay Deb, Central Queensland University, Australia
Javier Del Ser, Tecnalia Research & Innovation, Spain
Philippe Devienne, LIFL - Université Lille 1 - CNRS, France
Kamil Dimililer, Near East University, Cyprus
Martin Dobler, Vorarlberg University of Applied Sciences, Austria
Eugeni Dodonov, Intel Corporation- Brazil, Brazil
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Matthias Ehmann, Universität Bayreuth, Germany
Tarek El-Bawab, Jackson State University, USA
Nashwa Mamdouh El-Bendary, Arab Academy for Science, Technology, and Maritime Transport, Egypt
Mohamed Dafir El Kettani, ENSIAS - Université Mohammed V-Souissi, Morocco
Armando Ferro, University of the Basque Country (UPV/EHU), Spain
Anders Fongen, Norwegian Defence Research Establishment, Norway
Giancarlo Fortino, University of Calabria, Italy
Kary Främling, Aalto University, Finland
Steffen Fries, Siemens AG, Corporate Technology - Munich, Germany
Ivan Ganchev, University of Limerick, Ireland
Shang Gao, Zhongnan University of Economics and Law, China
Kamini Garg, University of Applied Sciences Southern Switzerland, Lugano, Switzerland
Rosario Giuseppe Garroppo, Dipartimento Ingegneria dell'informazione - Università di Pisa, Italy
Thierry Gayraud, LAAS-CNRS / Université de Toulouse / Université Paul Sabatier, France
Christos K. Georgiadis, University of Macedonia, Greece
Katja Gilly, Universidad Miguel Hernandez, Spain
Feliz Gouveia, Universidade Fernando Pessoa - Porto, Portugal
Kannan Govindan, Crash Avoidance Metrics Partnership (CAMP), USA
Bill Grosky, University of Michigan-Dearborn, USA
Vic Grout, Glyndŵr University, UK

Jason Gu, Singapore University of Technology and Design, Singapore
Christophe Guéret, Vrije Universiteit Amsterdam, Netherlands
Frederic Guidec, IRISA-UBS, Université de Bretagne-Sud, France
Bin Guo, Northwestern Polytechnical University, China
Gerhard Hancke, Royal Holloway / University of London, UK
Arthur Herzog, Technische Universität Darmstadt, Germany
Rattikorn Hewett, Whitacre College of Engineering, Texas Tech University, USA
Nicolas Hidalgo, Yahoo! Research Latin America, France
Quang Hieu Vu, EBTIC, Khalifa University, Arab Emirates
Hiroaki Higaki, Tokyo Denki University, Japan
Eva Hladká, Masaryk University, Czech Republic
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST), Korea
Anna Hristoskova, Ghent University - IBBT, Belgium
Ching-Hsien (Robert) Hsu, Chung Hua University, Taiwan
Christian Hübsch, Institute of Telematics, Karlsruhe Institute of Technology (KIT), Germany
Chi Hung, Tsinghua University, China
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Linda A. Jackson, Michigan State University, USA
Raj Jain, Washington University in St. Louis, USA
Edward Jaser, Princess Sumaya University for Technology - Amman, Jordan
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Georgios Kambourakis, University of the Aegean, Greece
Atsushi Kanai, Hosei University, Japan
Henrik Karstoft, Aarhus University, Denmark
Dimitrios Katsaros, University of Thessaly, Greece
Ayad ali Keshlaf, Newcastle University, UK
Reinhard Klemm, Avaya Labs Research, USA
Samad Kolahi, Unitec Institute Of Technology, New Zealand
Dmitry Korzun, Petrozavodsk State University, Russia / Aalto University, Finland
Evangelos Kranakis, Carleton University - Ottawa, Canada
Slawomir Kuklinski, Warsaw University of Technology, Poland
Andrew Kusiak, The University of Iowa, USA
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Frédéric Le Mouël, University of Lyon, INSA Lyon / INRIA, France
Nicolas Le Sommer, Université Européenne de Bretagne, France
Juong-Sik Lee, Nokia Research Center, USA
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Clement Leung, Hong Kong Baptist University, Hong Kong
Man-Sze Li, IC Focus, UK
Longzhuang Li, Texas A&M University-Corpus Christi, USA
Yaohang Li, Old Dominion University, USA
Jong Chern Lim, University College Dublin, Ireland
Lu Liu, University of Derby, UK
Damon Shing-Min Liu, National Chung Cheng University, Taiwan
Michael D. Logothetis, University of Patras, Greece
Malamati Louta, University of Western Macedonia, Greece

Maode Ma, Nanyang Technological University, Singapore
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Olaf Maennel, Loughborough University, UK
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Yong Man, KAIST (Korea advanced Institute of Science and Technology), South Korea
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Chengying Mao, Jiangxi University of Finance and Economics, China
Brandeis H. Marshall, Purdue University, USA
Sergio Martín Gutiérrez, UNED-Spanish University for Distance Education, Spain
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Hamid Mcheick, Université du Québec à Chicoutimi, Canada
Shawn McKee, University of Michigan, USA
Stephanie Meerkamm, Siemens AG in Erlangen, Germany
Kalogiannakis Michail, University of Crete, Greece
Peter Mikulecky, University of Hradec Kralove, Czech Republic
Moeiz Miraoui, Université du Québec/École de Technologie Supérieure - Montréal, Canada
Shahab Mokarizadeh, Royal Institute of Technology (KTH) - Stockholm, Sweden
Mario Montagud Climent, Polytechnic University of Valencia (UPV), Spain
Stefano Montanelli, Università degli Studi di Milano, Italy
Julius Müller, TU- Berlin, Germany
Juan Pedro Muñoz-Gea, Universidad Politécnica de Cartagena, Spain
Krishna Murthy, Global IT Solutions at Quintiles - Raleigh, USA
Alex Ng, University of Ballarat, Australia
Christopher Nguyen, Intel Corp, USA
Vlad Nicolici Georgescu, SP2 Solutions, France
Petros Nicopolitidis, Aristotle University of Thessaloniki, Greece
Carlo Nocentini, Università degli Studi di Firenze, Italy
Federica Paganelli, CNIT - Unit of Research at the University of Florence, Italy
Carlos E. Palau, Universidad Politecnica de Valencia, Spain
Matteo Palmonari, University of Milan-Bicocca, Italy
Ignazio Passero, University of Salerno, Italy
Serena Pastore, INAF - Astronomical Observatory of Padova, Italy
Fredrik Paulsson, Umeå University, Sweden
Rubem Pereira, Liverpool John Moores University, UK
Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada
Yulia Ponomarchuk, Far Eastern State Transport University, Russia
Jari Porras, Lappeenranta University of Technology, Finland
Neeli R. Prasad, Aalborg University, Denmark
Drogkaris Prokopios, University of the Aegean, Greece
Emanuel Puschita, Technical University of Cluj-Napoca, Romania
Lucia Rapanotti, The Open University, UK
Gianluca Reali, Università degli Studi di Perugia, Italy
Christoph Reinke, SICK AG, Germany
Jelena Revzina, Transport and Telecommunication Institute, Latvia
Karim Mohammed Rezaul, Glyndwr University, UK
Leon Reznik, Rochester Institute of Technology, USA

Joel Rodrigues, Instituto de Telecomunicações / University of Beira Interior, Portugal
Simon Pietro Romano, University of Napoli Federico II, Italy
Michele Ruta, Politecnico di Bari, Italy
Jorge Sá Silva, University of Coimbra, Portugal
Farzad Salim, Queensland University of Technology, Australia
Sébastien Salva, University of Auvergne, France
Ahmad Tajuddin Samsudin, Telekom Malaysia Research & Development, Malaysia
Josemaria Malgosa Sanahuja, Polytechnic University of Cartagena, Spain
Luis Enrique Sánchez Crespo, Sicaman Nuevas Tecnologías / University of Castilla-La Mancha, Spain
Paul Sant, University of Bedfordshire, UK
Brahmananda Sapkota, University of Twente, The Netherlands
Alberto Schaeffer-Filho, Lancaster University, UK
Peter Schartner, Klagenfurt University, System Security Group, Austria
Rainer Schmidt, Aalen University, Germany
Thomas C. Schmidt, HAW Hamburg, Germany
Didier Sebastien, University of Reunion Island, France
Zary Segall, Chair Professor, Royal Institute of Technology, Sweden
Dimitrios Serpanos, University of Patras and ISI/RC ATHENA, Greece
Jawwad A. Shamsi, FAST-National University of Computer and Emerging Sciences, Karachi, Pakistan
Michael Sheng, The University of Adelaide, Australia
Kazuhiko Shibuya, The Institute of Statistical Mathematics, Japan
Roman Y. Shtykh, Rakuten, Inc., Japan
Patrick Siarry, Université Paris 12 (LISi), France
Jose-Luis Sierra-Rodriguez, Complutense University of Madrid, Spain
Simone Silvestri, Sapienza University of Rome, Italy
Åsa Smedberg, Stockholm University, Sweden
Vasco N. G. J. Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Radosveta Sokullu, Ege University, Turkey
José Soler, Technical University of Denmark, Denmark
Boyeon Song, National Institute for Mathematical Sciences, Korea
Victor J. Sosa-Sosa, CINVESTAV-Tamaulipas, Mexico
Dora Souliou, National Technical University of Athens, Greece
João Paulo Sousa, Instituto Politécnico de Bragança, Portugal
Kostas Stamos, Computer Technology Institute & Press "Diophantus" / Technological Educational Institute of Patras, Greece
Vladimir Stantchev, SRH University Berlin, Germany
Tim Strayer, Raytheon BBN Technologies, USA
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Tae-Eung Sung, Korea Institute of Science and Technology Information (KISTI), Korea
Sayed Gholam Hassan Tabatabaei, Isfahan University of Technology, Iran
Yutaka Takahashi, Kyoto University, Japan
Yoshiaki Taniguchi, Osaka University, Japan
Nazif Cihan Tas, Siemens Corporation, Corporate Research and Technology, USA
Terje Jensen, Telenor Group Industrial Development / Norwegian University of Science and Technology, Norway
Alessandro Testa, University of Naples "Federico II" / Institute of High Performance Computing and Networking

(ICAR) of National Research Council (CNR), Italy
Stephanie Teufel, University of Fribourg, Switzerland
Parimala Thulasiraman, University of Manitoba, Canada
Pierre Tiako, Langston University, USA
Ioan Toma, STI Innsbruck/University Innsbruck, Austria
Orazio Tomarchio, Università di Catania, Italy
Kurt Tutschku, University of Vienna, Austria
Dominique Vaufreydaz, INRIA and Pierre Mendès-France University, France
Massimo Villari, University of Messina, Italy
Krzysztof Walkowiak, Wrocław University of Technology, Poland
MingXue Wang, Ericsson Ireland Research Lab, Ireland
Wenjing Wang, Blue Coat Systems, Inc., USA
Zhi-Hui Wang, School of Software, Dalian University of Technology, China
Matthias Wieland, Universität Stuttgart, Institute of Architecture of Application Systems (IAAS), Germany
Bernd E. Wolfinger, University of Hamburg, Germany
Chai Kiat Yeo, Nanyang Technological University, Singapore
Mark Yampolskiy, Vanderbilt University, USA
Abdulrahman Yarali, Murray State University, USA
Mehmet Erkan Yüksel, Istanbul University, Turkey

CONTENTS

pages: 65 - 83

Performance Evaluation of Distributed Mobile Application Virtualization Services

Chung-Ping Hung, Department of Electrical and Systems Engineering, Washington University in St. Louis, USA
Paul S. Min, Department of Electrical and Systems Engineering, Washington University in St. Louis, USA

pages: 84 - 94

Vision towards an Open Electronic Wallet on NFC Smartphones

Glenn Ergeerts, Artesis University College Antwerp, Belgium
Dries Schellekens, KU Leuven, Belgium
Frederik Schrooyen, Artesis University College Antwerp, Belgium
Rud Beyers, Artesis University College Antwerp, Belgium
Kevin De Kock, Artesis University College Antwerp, Belgium
Thierry Van Herck, Artesis University College Antwerp, Belgium

pages: 95 - 113

On-line Safety Monitor Based on a Safety Assessment Model and Hierarchical Deployment of a Multi-agent System

Amer Dheedn, Delmon University, Kingdom of Bahrain

pages: 114 - 125

Getting Anesthesia Online: The smartOR Network

Marcus Koeny, Chair for Medical Information Technology, RWTH Aachen University, Germany
Julia Benzko, Chair of Medical Engineering, RWTH Aachen University, Germany
Michael Czaplík, Department of Anesthesiology, University Hospital RWTH Aachen, Germany
Marian Walter, Chair for Medical Information Technology, RWTH Aachen University, Germany
Klaus Radermacher, Chair of Medical Engineering, RWTH Aachen University, Germany
Rolf Rossaint, Department of Anesthesiology, University Hospital RWTH Aachen, Germany
Steffen Leonhardt, Chair for Medical Information Technology, RWTH Aachen University, Germany

pages: 126 - 140

Hierarchical Routing for Small World Wireless Networks

Juhani Latvakoski, VTT Technical Research Centre of Finland, Finland

pages: 141 - 161

Design and Implementation of an Online XML Compressor for Large XML Files

Tomasz Muldner, Jodrey School of Computer Science Acadia University Wolfville, B4P 2A9 NS, Canada
Tyler Corbin, Jodrey School of Computer Science Acadia University Wolfville, B4P 2A9 NS, Canada
Jan Krzysztof Miziołek, IBI AL, University of Warsaw, Warsaw, Poland
Christopher Fry, Jodrey School of Computer Science Acadia University Wolfville, B4P 2A9 NS, Canada

pages: 162 - 172

Query-by-Appearance: Visual Query Expansion to Support Domain-Specific Retrieval of e-Books

Shuichi Kurabayashi, Keio University, Japan
Yuka Koike, Keio University, Japan

pages: 173 - 185

Performance Test Case Generation for Java and WSDL-based Web Services from MARTE

Antonio García-Domínguez, University of Cádiz, Spain

Inmaculada Medina-Bulo, University of Cádiz, Spain

Mariano Marcos-Bárcena, University of Cádiz, Spain

pages: 186 - 195

A Cross-Domain Query Navigation and Visualization System for Touchscreens that Exploits Social Search History

Shuichi Kurabayashi, Keio University, Japan

Ryo Shimaoka, Keio University, Japan

Performance Evaluation of Distributed Mobile Application Virtualization Services

Chung-Ping Hung* and Paul S. Min†

Department of Electrical and Systems Engineering, Washington University in St. Louis

One Brookings Drive, St. Louis, MO 63130, USA

Email: *chung23@wustl.edu †psm@wustl.edu

Abstract—In this paper, we first introduce how virtualization technologies can mitigate mobile application software publishing problems due to platform diversity and fragmentation. We propose a distributed server arrangement and the corresponding hand-off protocol to provide better user experience for application virtualization on mobile devices and evaluated the performance using the modified UMTS mobility models. We complete the establishment of quantitative relations between the performance improvement or impact and the infrastructure related parameters in the typical mobility model.

Keywords—telecommunication; wireless networks; computer networks; information technology; UMTS mobility model

I. INTRODUCTION

As computing devices get smaller, lighter, and more portable, computing becomes more focused on mobile applications. We expect this trend to continue for years to come.

Deploying application software on mobile computing devices can be a challenge for several reasons. First of all, various mobile operating systems exist and none is expected to dominate and set the standards for the mobile computing in ways the Windows operating systems have done for the desktop computing. Making a software program to be compatible with different mobile operating systems requires extra cost and effort – for example, developing on multiple SDKs (Software Development Kits).

Although compatibility across application-platforms also exist on typical desktop computers, it is far more difficult for mobile computing devices because of the additional constraints such as limited compute cycles in the mobile devices. Unlike operating systems for desktop computers, mobile operating systems are highly customized per product and secured against unauthorized user access. Generally, ordinary end-users cannot upgrade or patch their mobile operating systems to address application-platform compatibility issues, as can be done for the desktop computers. It is thus hinged upon software developers to provide compatibility across mobile platforms.

Virtualization can address the compatibility in deploying mobile application software on various mobile platforms. Theoretically, we can either use application streaming to deploy application software over the Internet and run the application software on top of a preinstalled runtime environment, i.e., virtual machine, or run the application software on a managed server while each client device deals with user inputs, such as keystrokes, and outputs, such as display updates from the

server [1], [2], [3]. We generally refer to the latter paradigm as the browser-based approach since web browsers provide an ideal framework for it.

Although technically plausible, deploying a virtual machine running on top of a mobile operating system provides an alternative way to distribute applications, which indeed violates the “Non-Compete” policy [4], [5] by marketplace operators¹. Consequently, the browser-based approach becomes the only practical way to provide application virtualization on mobile computing devices.

The conventional solution of web-based application virtualization involves setting up a server or a group of servers at a co-location center (or data center) provided by an Internet service provider (ISP). From the co-location center, application virtualization services are provided through the Internet. This configuration typically incurs long response latency and significantly reduces the user experience since every input must travel through a series of routers and bridges to the co-location center and the corresponding response has to traverse backward through a similar route. Each node along the route introduces processing delay, queuing delay, and transmission delay.

To alleviate this issue, we propose an alternative configuration which partitions a service area into multiple smaller service areas with own server(s) [7]. The proposed configurations can significantly reduce delays since each server is closer to its user.

The proposed configuration, however, has to handle hand-off, i.e., mobile stations moving from one service area to another. We also propose a hand-off protocol offering seamless user experience in Section IV.

The proposed configuration comes with a price, such as introducing longer response latency during hand-offs. We use an analytical approach to evaluate the performance as a result of infrastructure arrangement [7].

We further set up a simulation environment based on the

¹VMware’s Mobile Virtualization Platform (MVP) [6], which implements this paradigm, is not available in Android Marketplace. To install MVP on an Android phone requires sideloading, and only Android platform leaves this loophole to install apps outside the marketplace, which is at the mercy of Google and wireless service providers. In fact, some wireless providers do block sideloading on some Android phones. Furthermore, among the major mobile device players, only Android is supported by MVP. Therefore, even VMware starts their own app store for MVP, it does not help cross-platform software deployment anyway.

UMTS urban pedestrian model and vehicular mobility model, and use the empirical approach to establish the correlations between the performance and the size of local service areas [8], [9].

II. RELATED WORK

There are several papers proposed to optimize service migration though for different applications. Bienkowski et al. proposed competitive analysis for service migration in optimizing the server allocation in VNets in [10]. Arora et al. proposed some strategies for flexible server allocation in [11] following the previous work [10]. Although these works were not specifically for mobile application virtualization, they provide a precious insight on the performance evaluation for dynamic service allocation considering both user experience and operational cost. However, the analytical approach used in these works is topological and does not focus on the user mobility and interaction models. Furthermore, the authors of [10] and [11] allow services being temporarily interrupted during migrations, which is not feasible for application virtualization services. In the proposed configuration, application services are available to users with reduced performance during hand-offs.

III. DISTRIBUTED APPLICATION VIRTUALIZATION SERVICE CONFIGURATION

Running application software on a remote server is conceptually similar to the usage model of time-sharing mainframe computers in the 1960s [12]. Although the communication bandwidth between terminals and mainframe servers at that time was low by the recent standards, it did not affect the user experience thanks to the text-only display and short traverse distance. However, in recent application virtualization technologies such as Virtual Desktop Infrastructure (VDI) proposed by VMware [13], much more complex and bloated content must be exchanged over much longer distances between clients and servers, especially for mobile users.

An infrastructure ready to offer mobile users application virtualization services includes base stations (BSs) covering the whole service area, a core network connecting base stations and servers together, and a server hosting the services. A command sent by a mobile station (MS) has to travel over the wireless channel to the BS, go through the backhaul network to the server, and then make some changes on the server. Should any update corresponding to the command be sent to the MS, the information has to travel all the way backward. In order to reduce the network delay generated by long transmission distances among the backhaul network, we deploy multiple servers among a wide area to serve their nearby MSs in the proposed configuration, instead of setting up a group of servers located at one data center serving all MSs.

In the proposed configuration, each server connects to several nearby BSs which form a local service group (LSG). The area covered by the BSs of the same LSG is defined as the local service area (LSA). Every BS belongs to one LSG in order to provide the service over the wireless network's

coverage area. When a user demands a virtual application program, the server of the LSG, based on VDI [13] paradigm, starts a virtual machine (VM) dedicated to the user and launches the application software on top of it. The MS only handles inputs and outputs that interact with the VM at the server.

As long as the MS stays in the same LSA, the user can enjoy using application software with low response latency. If the MS moves from the original LSA to a nearby one, a hand-off at the VM level, which transfers the runtime environment to the server of the next LSG, is triggered. The detail of the hand-off protocol will be proposed in the next section.

IV. HAND-OFF PROTOCOL

The purpose of the proposed hand-off protocol is to transfer minimum information required to recreate the runtime environment on a remote server, i.e., the snapshot, without interrupting the service. No matter how small the snapshot is, it still takes a period of time before the next server receives the complete snapshot and is ready to take over the service. In order to provide a seamless user experience during this period, the next server has to record all inputs from the MS, relay all inputs to the previous server, and relay all output from the previous server to the MS, until the runtime environment resumes locally. The proposed hand-off protocol is described as below:

- 1) When an MS moves from Server A's to Server B's LSA and sends an input command, Server B notices a newcomer within its LSA.
- 2) Server B broadcasts the newcomer's identification to all geographically nearby servers.
- 3) Server A, which hosts the MS's runtime environment, i.e., its VM server, responds Server B's inquiry. Now Server B knows the newcomer's VM server is Server A.
- 4) Server B records and relays the user's input commands to Server A, signals Server A to transfer the runtime environment, and relays display updates from Server A to the newcomer.
- 5) Once Server A is signaled to transfer the runtime environment, it takes a snapshot.
- 6) Besides continually responding to the input commands relayed from Server B as the MS is still in its LSA, Server A also sends the snapshot to Server B in the background.
- 7) Once Server B receives the complete snapshot and recreates the runtime environment from the snapshot and base data, it internally feeds the input queue, which was recorded during the transition period, to the runtime environment. Therefore, the runtime environment state on Server B is synchronous with that on Server A after the snapshot was transferred.
- 8) Server A completely stops serving the MS, the MS's VM server is now Server B instead.

The timeline of the proposed hand-off protocol is illustrated in Figure 1.

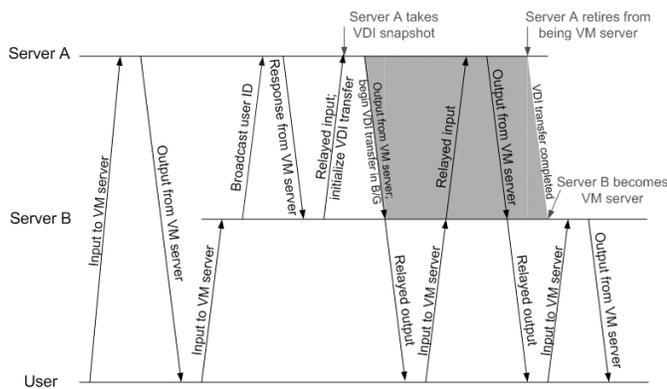


Fig. 1. Protocol timeline for a mobile station moving from Server A to Server B.

If the MS turns around and reenters Server A's LSA before the hand-off is completed, Server A can preempt the snapshot transmission and resume serving the MS as if the hand-off never happened. Since Server B relays all inputs to Server A while the MS is absent from Server A's LSA, aborting the hand-off procedure would not generate any glitch noticed by the user. This hand-off abortion mechanism can prevent unnecessary data transmission from moving VM servers back and forth if an MS were moving around the edge of an LSA.

On the other hand, if the MS moves to Server C's LSA before the hand-off is completed, Server C initializes another hand-off procedure with Server B. In addition to the snapshot, Server B has to transfer the input record before Server C joins the hand-off chain. We allow pipelining transmission to reduce hand-off periods and shorten subsequent hand-off chains in this scenario.

V. PERFORMANCE EVALUATION USING FREE PARTICLE MOBILITY MODEL

We define the response time as the average time interval between a user sends an input and gets an expected output update. The proposed server configuration is meant to improve the response time by reducing propagation delay along the communication route between each base station to the server which is hosting the service. Factors other than the propagation delay, such as wireless communication technologies and computational capabilities provided by servers, would affect the user experience and the quality of our service. Most of them, however, either affect both configurations equally, or can be overcome with reasonable cost.

The proposed configuration reduces the propagation delay and thus provides more responsive user experiences when users are standing still. When a hand-off occurs, however, the user may experience longer response time waiting for the information to be exchanged between two servers before the runtime environment is successfully taken over by the new server. The smaller each local service area is, the higher occurrence probability of hand-offs the user may experience. Therefore, we have to quantitatively estimate and compare the

propagation delays of the conventional and the proposed server configurations.

The precise propagation delay analysis depends on a wireless service provider's core network topology and its users' moving pattern record. Instead of acquiring those field data, we focus on the intrinsic properties of the two configurations. There are two approaches to estimate the average response time due to propagation delay; one assumes continuous service areas, the other is based on the optimal arrangement of base stations. The details are presented in the following subsections.

A. Continuous Service Area Approach

In this approach, we simplify the communication model between mobile stations and servers. Here are our assumptions:

- 1) The whole service area can be covered by a single server, or proximately by multiple servers, each having a regular hexagon shaped service area seamlessly tiled together as a service array.
- 2) A mobile station can directly communicate with the server everywhere in its (local) service area.
- 3) Users are uniformly distributed geographically in the beginning. Users can either move a certain distance in any direction, or stay at the same location for a while.
- 4) The propagation delay of each link is proportional to its length.
- 5) Each server's allocation is geographically optimized, that is, each server is located at the center of its (local) service area to reduce the average propagation delay.

The traverse time in our case is defined by:

$$T_{traverse} = 2 \cdot (1 - P_{HO}) \cdot (T_r + T_l) + 2 \cdot P_{HO} \cdot T_{HO} \quad (1)$$

where P_{HO} is the probability of transactions which either trigger hand-offs or occur during each handoff, T_r is the radio propagation delay, T_l is the line propagation delay, and T_{HO} is the prolonged traverse time during each hand-off according to the proposed protocol. To simplify the problem, we only compare the following three configurations covering the same amount of area:

- A A single server covering a regular hexagon service area of edge length L .
- B 7 servers, each covering a regular hexagon service area of edge length $\frac{L}{\sqrt{7}}$, as shown in Figure 2.
- C 12 servers, each covering a regular hexagon service area of edge length $\frac{L}{\sqrt{12}}$, as shown in Figure 3.

1) *Average Transmission Distance:* We can calculate the distance from an arbitrary point within each hexagon-shaped service area to the center of the area, where the optimal server is. Since we assume that our users' locations are uniformly distributed geographically in our service area, we can estimate the average transmission distance for each user in terms of edge length of the service area. Due to the symmetry of hexagons, the average distance from an arbitrary point within a hexagon to its center is equivalent to the average distance from an arbitrary point within a 30-60-90 triangle to the 30-degree vertex as shown in Figure 4.

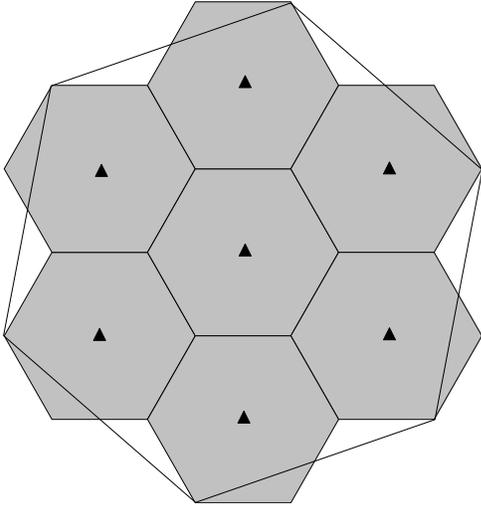


Fig. 2. Service area of 7-server configuration compares with of single-server one.

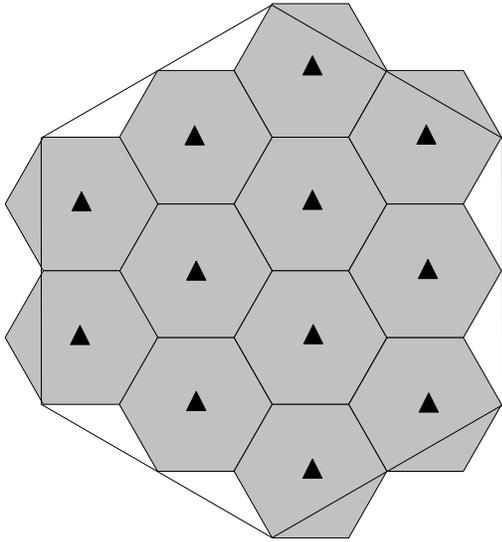


Fig. 3. Service area of 12-server configuration compares with of single-server one.

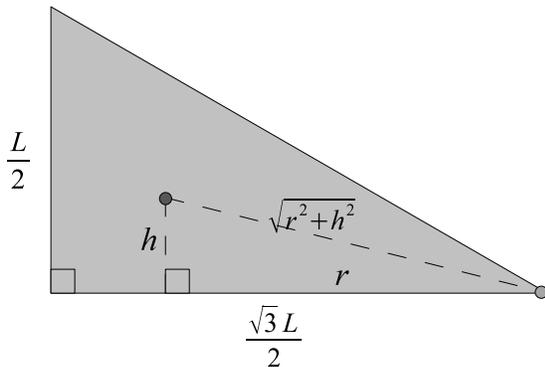


Fig. 4. 30-60-90 triangle as part of hexagon with edge length L , used to estimate average distance to the lower right vertex.

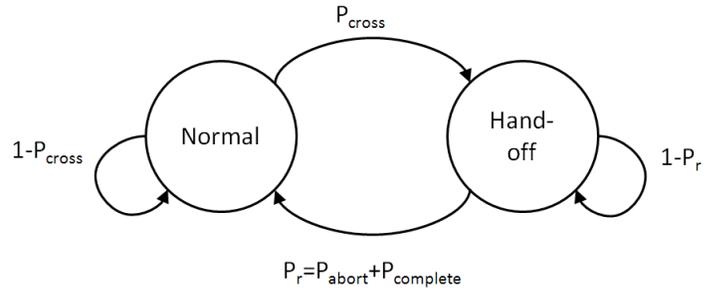


Fig. 5. The Markov chain of a moving MS's status.

By integrating $\sqrt{r^2 + h^2}$ along h and r , as shown below:

$$\begin{aligned}
 & \int_0^{\frac{\sqrt{3}L}{2}} \int_0^{\frac{r}{\sqrt{3}}} \left\{ \sqrt{r^2 + h^2} \right\} dh dr \\
 &= \int_0^{\frac{\sqrt{3}L}{2}} \left\{ \frac{h}{2} \sqrt{r^2 + h^2} + \frac{r^2}{2} \ln \left| h + \sqrt{r^2 + h^2} \right| \right\} \Big|_0^{\frac{r}{\sqrt{3}}} dr \\
 &= \int_0^{\frac{\sqrt{3}L}{2}} r^2 dr \cdot \left\{ \frac{1}{3} + \frac{\ln(3)}{4} \right\} \\
 &= \frac{\sqrt{3}L^3}{8} \cdot \left\{ \frac{1}{3} + \frac{\ln(3)}{4} \right\} \quad (2)
 \end{aligned}$$

By averaging the result above by the whole triangle area, the average transmission distance for each user in terms of the edge length of the service area is:

$$\frac{\frac{\sqrt{3}L^3}{8} \cdot \left\{ \frac{1}{3} + \frac{\ln(3)}{4} \right\}}{\frac{\sqrt{3}L^2}{8}} = \left\{ \frac{1}{3} + \frac{\ln(3)}{4} \right\} \cdot L \approx 0.60799L \quad (3)$$

2) *Probability of Transactions Relevant to Hand-off*: The transition between normal and hand-off mode of each moving MS can be represented by a simple two-state Markov chain as shown in Figure 5.

As we can see in Figure 5, a moving MS in the normal state gets into the hand-off state when it moves across the border of its current local service area with probability P_{cross} . On the other hand, a moving MS in the hand-off state can go back to the normal state either by completing the hand-off procedure with probability $P_{complete}$, or by returning to the previous local service area and preempting the hand-off procedure with probability P_{abort} . The summation of $P_{complete}$ and P_{abort} is P_r , which represents the total probability for a moving MS in the hand-off state to return to the normal state.

By steady-state analysis, we can derive the probability of transactions relevant to hand-offs, i.e., P_{HO} , as below:

$$\begin{aligned}
 & [1 - P_{HO} \quad P_{HO}] \begin{bmatrix} 1 - P_{cross} & P_{cross} \\ P_r & 1 - P_r \end{bmatrix} \\
 &= [1 - P_{HO} \quad P_{HO}] \\
 & P_{HO} = \frac{P_{cross}}{P_{cross} + P_r} \leq \frac{P_{cross}}{P_{cross} + P_{complete}} \quad (4)
 \end{aligned}$$

As we can see, the two factors P_{cross} and P_r affect P_{HO} . Both factors depend on users' mobility and the dimension of the service areas. Assume the average moving speed of an MS

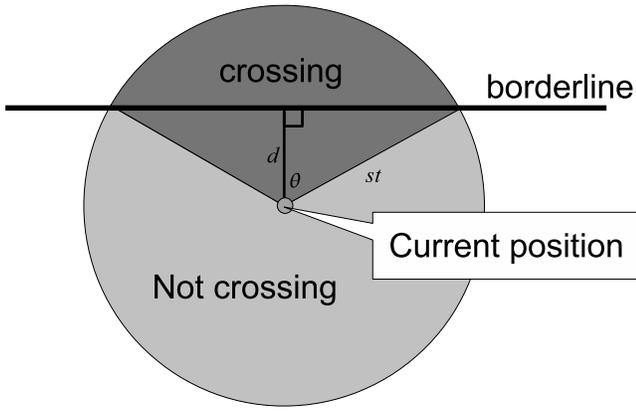


Fig. 6. For an MS close to the borderline who can freely choose its direction, the probability of crossing the borderline in the next time instance is $\frac{2\theta}{2\pi}$.

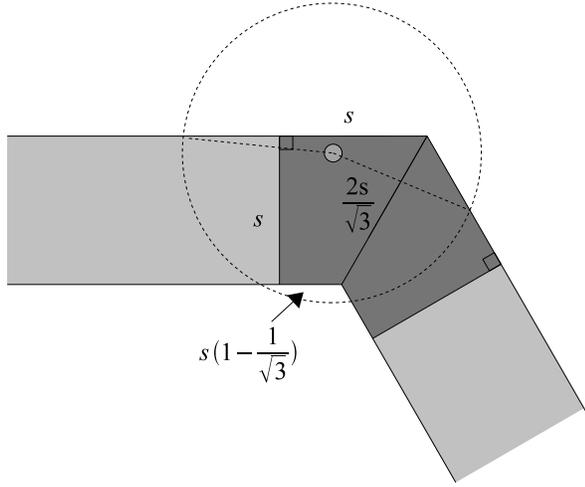


Fig. 7. Users at the singular area (dark area) have higher P_{cross} ; the above equation can only apply in the normal areas (light areas).

is $\frac{s}{\Delta t}$. To make it possible to trigger a hand-off in the following time instance Δt , the MS has to be within s from the current service area's borderline. Furthermore, the probability of an MS satisfying this prerequisite actually crossing the borderline and thus triggers a hand-off depends on how close to the borderline it is as shown in Figure 6.

Therefore, the probability of an MS which is located d from the borderline with speed $\frac{s}{\Delta t}$ actually crossing the borderline in the next time instance Δt is given by:

$$P_{cross}(d, s) = \begin{cases} \frac{1}{\pi} \cdot \cos^{-1}\left(\frac{d}{s}\right) & 0 \leq d \leq s \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

However, since the service areas are hexagon-shaped, the borderline within the moving range is not always a straight line. As shown in Figure 7, the above equation does not apply to the MS located in the *singular area*, i.e., the area near vertices. It is so complex to estimate exact P_{cross} at singular area, such that we only calculate the range of P_{cross} instead.

We define $\hat{P}_{cross}(d, s)$ as the probability of an MS in the singular area crossing the borderline. Intuitively, the upper bound of $\hat{P}_{cross}(d, s)$ is $\frac{2}{3}$, in case of the MS starting at the

corner, while the lower bound is $P_{cross}(d, s)$. The singular area would not be a problem in our estimation if L is relatively larger than s .

For each hexagon-shaped service area, the probability for an arbitrary MS crossing the borderline and triggering a hand-off is:

$$\begin{aligned} \bar{P}_{cross}(L, s, n) &= \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} + \frac{2ns^2(2\sqrt{3}-1) \cdot \hat{P}_{cross}}{9L^2} \end{aligned} \quad (6)$$

where \hat{P}_{cross} is the average probability of an MS in the singular area crossing the borderline, and n is the number of edges which border another service area. The detail derivation will be presented in Appendix A.

Since $\hat{P}_{cross} \leq \frac{2}{3}$,

$$\begin{aligned} \bar{P}_{cross}(L, s, n) &\leq \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} + \frac{2ns^2(2\sqrt{3}-1) \cdot \frac{2}{3}}{9L^2} \\ &= \frac{2ns\{3\sqrt{3}L + 2(2\sqrt{3}\pi - 2\pi - 3\sqrt{3})s\}}{27\pi L^2} \\ &= \frac{2\sqrt{3}n}{9\pi} \cdot \left(\frac{s}{L}\right) \\ &\quad + \frac{4(2\sqrt{3}\pi - 2\pi - 3\sqrt{3})n}{27\pi} \cdot \left(\frac{s}{L}\right)^2 \end{aligned} \quad (7)$$

For $s \ll L$,

$$\bar{P}_{cross}(L, s, n) \approx \frac{2\sqrt{3}n}{9\pi} \cdot \left(\frac{s}{L}\right) \quad (8)$$

In the proposed design, the hand-off procedure takes a period of time while the MS can continue sending requests. The duration of a complete hand-off $T_{complete}$, as a result of the total amount of data which are transferred for each hand-off D_{sync} , the transmission bandwidth provided by the link between the two adjacent servers BW_s , and the transmission latency of the link T_{ls} , all affect the fraction of transactions relevant to hand-offs. The equation is given by:

$$T_{complete} = \frac{D_{sync}}{BW_s} + T_{ls} \quad (9)$$

Once a user triggers a hand-off, the subsequent requests within $T_{complete}$ are categorized as hand-off related transactions. In other words, there are at least $P_{complete} = \frac{1}{T_{complete}}$ of MSs in the hand-off status return to the normal status in average. The actual rate of leaving the hand-off status P_r should be substantially higher since some MSs preempt the hand-off. However, the hand-off abortion rate P_{abort} is very difficult to be derived with analytical approaches. Consequently, we take $P_{complete}$ as a reference of P_r first and discuss the relation between them later.

Now we can derive P_{HO} by the following equation:

$$\begin{aligned}
P_{HO} &= \frac{\bar{P}_{cross}}{\bar{P}_{cross} + \frac{\alpha}{T_{complete}}} \\
&= \frac{\bar{P}_{cross}}{\bar{P}_{cross} + \frac{\alpha}{\frac{D_{sync}}{BW_s} + T_{ls}}} \\
&= \frac{\frac{2\sqrt{3}E(n)}{9\pi} \cdot \left(\frac{s}{L}\right)}{\frac{2\sqrt{3}E(n)}{9\pi} \cdot \left(\frac{s}{L}\right) + \frac{\alpha}{\frac{D_{sync}}{BW_s} + T_{ls}}} \\
&= \frac{2\sqrt{3}E(n) \left(\frac{s}{L}\right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls} \right\}}{2\sqrt{3}E(n) \left(\frac{s}{L}\right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls} \right\} + 9\pi\alpha} \quad (10)
\end{aligned}$$

where $\alpha = \frac{P_r}{P_{complete}} > 1$ is the average hand-off duration, and $E(n)$ is the average number of edges which border another service area, which is 0, $\frac{24}{7}$, and 4 for Configuration A, B, and C, respectively.

We can roughly conclude that P_{HO} can be increased by higher MS mobility, a larger volume of the data required for the synchronization, and a longer transmission latency between the servers. On the other hand, it will be reduced by a wider service area, a higher bandwidth between the servers, and a higher rate of hand-off abortion. However, the transmission latency between the two adjacent servers is proportional to the service range. We will see how the service range affects the average response time in the following subsection.

3) *Average Response Time Comparison of the Three Configurations:* In Configuration A, there is only one server thus no hand-off mechanism. The average traverse time of Configuration A is quite straightforward:

$$T_{traverse}^A = 2 \cdot (T_r + T_l^A) \quad (11)$$

Now we have to consider hand-offs in Configuration B. Its average traverse time is:

$$\begin{aligned}
T_{traverse}^B &= 2 \cdot (1 - P_{HO}^B) \cdot (T_r + T_l^B) + 2 \cdot P_{HO}^B \cdot T_{HO}^B \\
&= 2 \cdot (1 - P_{HO}^B) \cdot \left(T_r + \frac{T_l}{\sqrt{7}}\right) \\
&\quad + 2 \cdot P_{HO}^B \cdot (T_r + T_{lmax} + T_{ls}) \\
&= 2 \left\{ T_r + \frac{T_l}{\sqrt{7}} - \frac{P_{HO}^B T_l}{\sqrt{7}} + \frac{\left\{ \frac{1}{2} + \frac{3\ln(3)}{4} + \sqrt{3} \right\} P_{HO}^B T_l}{\sqrt{7} \left(\frac{1}{3} + \frac{\ln(3)}{4} \right)} \right\} \\
&= 2 \left\{ T_r + \frac{T_l}{\sqrt{7}} \left[1 + P_{HO}^B \left\{ \frac{12\sqrt{3} + 2 + 6\ln(3)}{4 + 3\ln(3)} \right\} \right] \right\} \quad (12)
\end{aligned}$$

where T_{lmax} is the propagation delay between the MS and the new server during hand-offs, which is $\left\{ \frac{1}{2\sqrt{3}} + \frac{3\ln(3)}{8\sqrt{3}} \right\} T_{ls}$, since we assume that the MSs are still located around the borderline at the time. The detail derivation, which is very similar to 2, will be presented in Appendix B.

Therefore, if we expect that Configuration B would outperform Configuration A, i.e., $T_{traverse}^B < T_{traverse}^A$, we can

estimate the upper bound of P_{HO}^B as below:

$$1 + P_{HO}^B \left\{ \frac{12\sqrt{3} + 2 + 6\ln(3)}{4 + 3\ln(3)} \right\} < \sqrt{7}$$

$$P_{HO}^B < \frac{(\sqrt{7} - 1)(4 + 3\ln(3))}{12\sqrt{3} + 2 + 6\ln(3)} \approx 0.4087356087 \quad (13)$$

This constrain is generally considered very slack.

Similarly, the average traverse time for Configuration C is:

$$\begin{aligned}
T_{traverse}^C &= 2 \left\{ T_r + \frac{T_l}{\sqrt{12}} \left[1 + P_{HO}^C \left\{ \frac{12\sqrt{3} + 2 + 6\ln(3)}{4 + 3\ln(3)} \right\} \right] \right\} \quad (14)
\end{aligned}$$

And the upper bound of P_{HO}^C to outperform Configuration A is:

$$1 + P_{HO}^C \left\{ \frac{12\sqrt{3} + 2 + 6\ln(3)}{4 + 3\ln(3)} \right\} < \sqrt{12}$$

$$P_{HO}^C < \frac{(\sqrt{12} - 1)(4 + 3\ln(3))}{12\sqrt{3} + 2 + 6\ln(3)} \approx 0.6119795056 \quad (15)$$

Furthermore, to outperform Configuration B given the same BW_s , the criteria is:

$$\begin{aligned}
&\frac{T_l}{\sqrt{12}} \left\{ 1 + P_{HO}^C \left\{ \frac{12\sqrt{3} + 2 + 6\ln(3)}{4 + 3\ln(3)} \right\} \right\} \\
&< \frac{T_l}{\sqrt{7}} \left\{ 1 + P_{HO}^B \left\{ \frac{12\sqrt{3} + 2 + 6\ln(3)}{4 + 3\ln(3)} \right\} \right\} \\
&\Rightarrow \frac{8\sqrt{21} \left(\frac{\sqrt{12}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + \sqrt{\frac{7}{12}} T_{ls}^B \right\}}{8\sqrt{3} \left(\frac{\sqrt{12}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + \sqrt{\frac{7}{12}} T_{ls}^B \right\} + 9\pi\alpha_C} \\
&\quad - \frac{\frac{96}{7} \left(\frac{\sqrt{7}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^B \right\}}{\frac{16\sqrt{3}}{7} \left(\frac{\sqrt{7}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^B \right\} + 3\pi\alpha_B} \\
&< \frac{(\sqrt{12} - \sqrt{7})(4 + 3\ln(3))}{12\sqrt{3} + 2 + 6\ln(3)} \quad (16)
\end{aligned}$$

The detail derivation will be presented in Appendix C.

The inequality above provides an accurate bound of the function of $\frac{s}{L}$, $\frac{D_{sync}}{BW_s}$, T_{ls}^B , α_B , and α_C . It is, however, too complex to help us to determine which configuration is better given a set of system parameters. Fortunately, we can discover the benefit brought by a more distributed infrastructure arrangement by simplify the inequality above based on sensible approximations. First of all, in most case T_{ls} is negligible comparing to $\frac{D_{sync}}{BW_s}$. Therefore, we can replace all $T_{complete}$ by $\frac{D_{sync}}{BW_s}$. Secondly, as $s \ll L$, P_{abort} 's in both configurations are approximately the same. In consequence, $\alpha_B \approx \alpha_C$. Therefore, we can further set $\alpha P_r = \beta P_{cross}^B$ where $\beta > 0$ to simplify the inequality.

Since

$$\hat{P}_{cross}(L, s) \approx \frac{2\sqrt{3}E(n)}{9\pi} \cdot \left(\frac{s}{L}\right) \quad (17)$$

Therefore,

$$\begin{aligned} \frac{P_{cross}^C}{P_{cross}^B} &= \frac{2\sqrt{3}\cdot 4}{9\pi} \cdot \left(\frac{\sqrt{12}s}{L} \right) \\ &= \frac{2\sqrt{3}\cdot \frac{24}{7}}{9\pi} \cdot \frac{\sqrt{7}s}{L} \\ \Rightarrow P_{cross}^C &= \sqrt{\frac{7}{3}} P_{cross}^B \end{aligned} \quad (18)$$

Now we can represent P_{HO}^B and P_{HO}^C only in terms of P_{cross}^B and β :

$$\begin{aligned} P_{HO}^B &= \frac{P_{cross}^B}{P_{cross}^B + \alpha P_r} = \frac{P_{cross}^B}{P_{cross}^B + \beta P_{cross}^B} \\ &= \frac{1}{1 + \beta} \\ P_{HO}^C &= \frac{P_{cross}^C}{P_{cross}^C + \alpha P_r} = \frac{\sqrt{\frac{7}{3}} P_{cross}^B}{\sqrt{\frac{7}{3}} P_{cross}^B + \beta P_{cross}^B} \\ &= \frac{\sqrt{7}}{\sqrt{7} + \sqrt{3}\beta} \end{aligned} \quad (19)$$

And then rewrite the inequality in terms of β :

$$\begin{aligned} \sqrt{7}P_{HO}^C - \sqrt{12}P_{HO}^B &< \frac{(\sqrt{12} - \sqrt{7})(4 + 3\ln(3))}{12\sqrt{3} + 2 + 6\ln(3)} \\ \frac{7}{\sqrt{7} + \sqrt{3}\beta} - \frac{2\sqrt{3}}{1 + \beta} &< \frac{(\sqrt{12} - \sqrt{7})(4 + 3\ln(3))}{12\sqrt{3} + 2 + 6\ln(3)} \\ \frac{(7 - 2\sqrt{21}) + \beta}{\sqrt{7} + (\sqrt{7} + \sqrt{3})\beta + \sqrt{3}\beta^2} &< \frac{(\sqrt{12} - \sqrt{7})(4 + 3\ln(3))}{12\sqrt{3} + 2 + 6\ln(3)} \\ &\Rightarrow 0.352\beta^2 - 0.110\beta + 2.703 > 0 \end{aligned} \quad (20)$$

which is true for all β .

Therefore, we can conclude that if $s \ll L$, $\frac{D_{sync}}{BW_s} \gg T_{ls}$, and $\alpha_B \approx \alpha_C$, Configuration C can always outperform Configuration B in terms of traverse delay.

4) *The Actual Rate of Leaving Hand-off State*: To better understand P_r and its relation to $T_{complete}$, we wrote a simple simulation program to empirically measure the average time an MS stays in the hand-off status. The program simulates an MS originally located close to a borderline, whose distance to it is uniformly distributed from 0 to s . Before it cross the borderline and triggers a hand-off, it randomly choose a direction from $-\pi$ to π and step forward s , which ensures it either crosses, or approaches to, the borderline. Once it triggers a hand-off, it can randomly choose any direction to step forward until the predetermined $T_{complete}$ runs out or it moves back to the other side of the borderline. The time each MS stays in the hand-off status is gauged and averaged in the end of the program.

The average time MSs stay in the hand-off status given different $T_{complete}$ is shown in Figure 8.

The value of α , which varies in a similar curve as in Figure 8 is shown in Figure 9.

As we can see in Figure 8 and Figure 9, the actual rate of leaving hand-off state P_r , which is the inverse of actual average hand-off duration, only fluctuates slightly in response

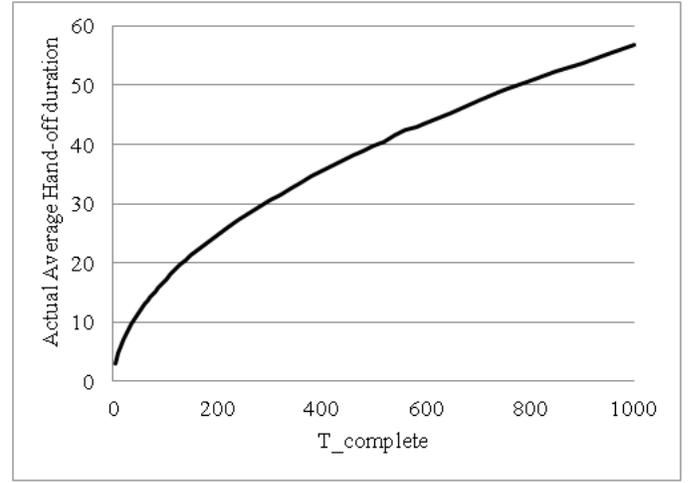


Fig. 8. The actual average hand-off duration.

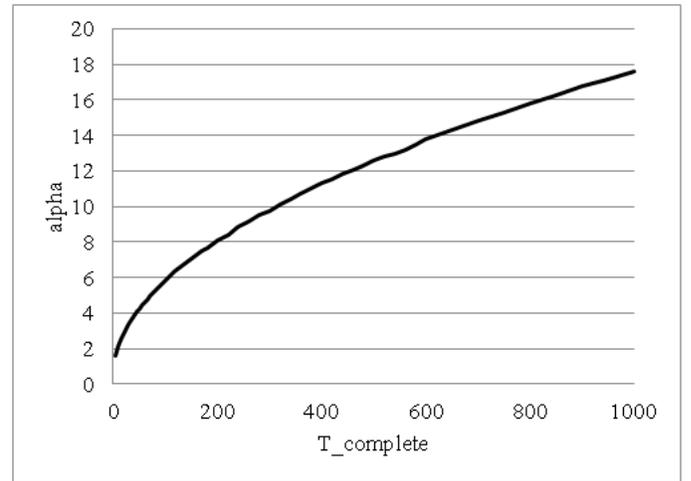


Fig. 9. The value of α given different $T_{complete}$.

to $T_{complete}$. Therefore, we can assure that ignoring T_{ls} and subsequently assuming that $T_{complete}$'s are identical for both configurations are sensible.

B. Optimal Arranged Base Stations Approach

In this approach, the service area is covered by a group of base stations, each connected to a server. Unlike the continuous service area approach which assumes each service area is a perfect regular hexagon, in this model the service areas are shaped by overlapping disks, each covered by a base station with omni-directional antenna. Consequently, each (local) service area is similar to a regular hexagon but with some "ripples" around the edges, which make it very difficult to estimate the hand-off probability. We can, however, proximately estimate it in certain conditions. Here are the assumptions, which are slightly different from those of the other approach:

- 1) The whole service area is covered by minimum number of base stations with omni-directional antennae. In other

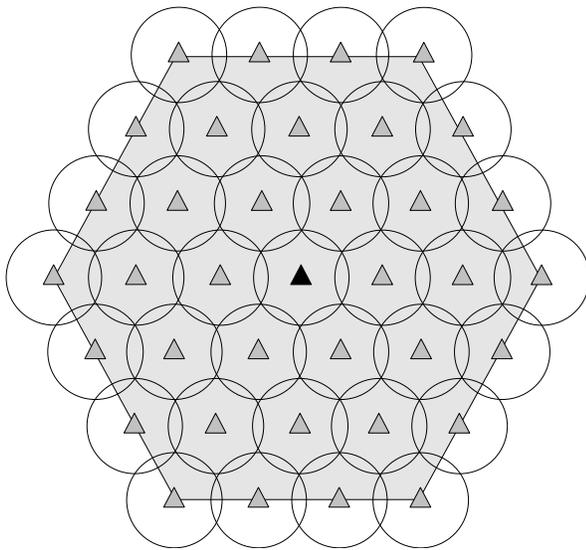


Fig. 10. Service area of single-server configuration with $m = 3$.

words, base stations are located at unit points of a two-dimensional Synergetics coordinates [14].

- 2) We can either connect all base stations to one server, or separate base stations into several groups and connect them to the server of each group. The optimal service area of each group is approximately a regular hexagon.
- 3) Users are uniformly distributed geographically in the beginning. Users can either move a certain distance in any direction, or stay at the same location for a while.
- 4) The propagation delay of each link is proportional to its length.
- 5) Each server's allocation is geographically optimized, that is, each server is located in the center of its (local) service area to reduce average propagation delay. The traverse time in our service is defined by (1) as well.

Again, we compare the following two configurations covering the same area.

- A: $(3m^2 + 3m + 1)$ base stations are placed like a regular hexagon, where m is the number of the base stations' intervals along one of the hexagon's edges. Each interval is $\sqrt{3}R$ long, where R is the effective communication range of each base station. An example is illustrated in Figure 10.
- B: 7 servers, each connected to $(3\lceil \frac{m}{3} \rceil^2 + 3\lceil \frac{m}{3} \rceil + 1)$ base stations as a local service area. The base stations in each local service area are placed like a regular hexagon with $\lceil \frac{m}{3} \rceil$ intervals along one of its edge, as shown in Figure 11.

1) *Average Transmission Distance*: The average transmission distance in this approach is the discrete version of the continuous service area's counterpart. However, it is very difficult to represent in terms of m , as shown below:

$$\frac{3r \sum_{t=0}^{m-1} \sum_{k=1}^{m-t} \sqrt{3(2k+t)^2 + 9t^2}}{3m^2 + 3m + 1} \quad (21)$$

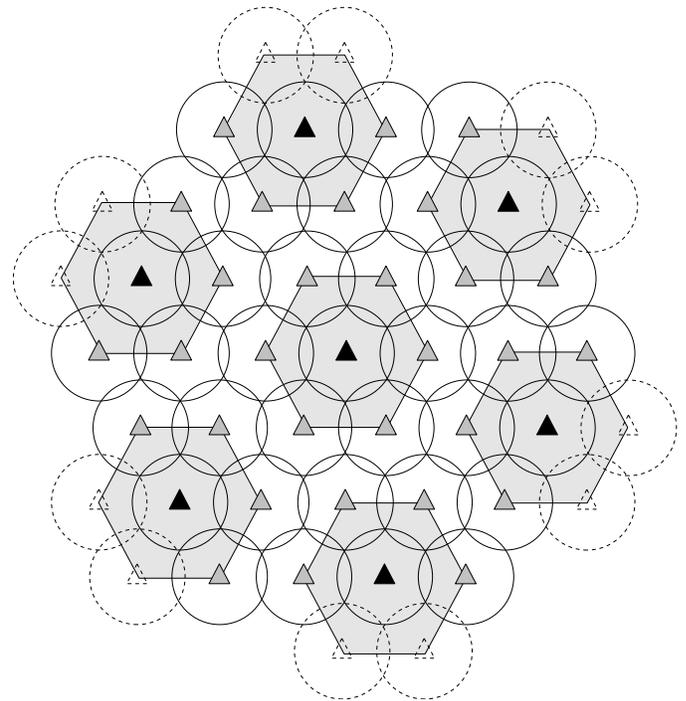


Fig. 11. Service areas of 7-server configuration, each with $m = 1$, covering the same amount of area.

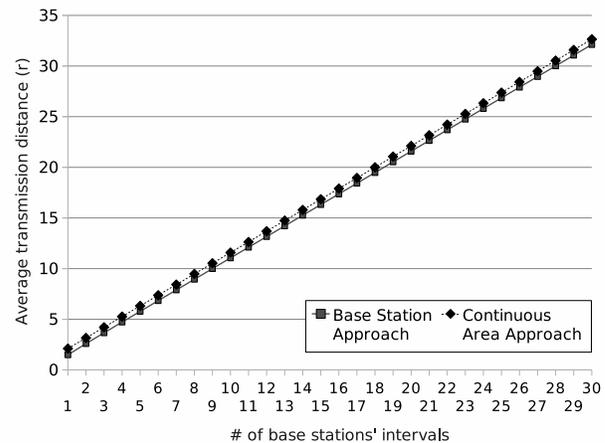


Fig. 12. Comparison of average transmission distances of different approaches covering approximately equal service area.

Fortunately, we find out that the average transmission distance in this approach is approximately linear and gets closer to its continuous counterpart as m increases according to the computer calculation, as shown in Figure 12.

In other words, we can estimate the average transmission distance by either (21), or the continuous counterpart (3) with comparable parameters. In the later sections, we will use the latter one to focus on the quantitative relationships between the parameters and the performance rather than the exact value.

2) *Probability of Transactions Relevant to Hand-offs*: Due to the irregular shape of each local service area, it is difficult to estimate the exact probability of an arbitrary user around the

border moving out of the service area by equations. However, if users' moving distances in each time instance are relatively short compared to a base station's effective communication range, the perimeter of each local service area at any point is near a straight line from a user's point of view.

Therefore, we can borrow the results from the continuous counterpart (6) to estimate the probability of a user crossing the borderline. The average probability of a user crossing the borderline along an arbitrary line which is perpendicular to the assumed straight borderline is:

$$\begin{aligned}\delta\bar{P}_{cross}(s) &= \int_0^s \left\{ \frac{1}{\pi} \cos^{-1} \left(\frac{d}{s} \right) \right\} dd \\ &= \frac{s}{\pi} \int_0^1 \cos^{-1}(k) dk \\ &= \frac{s}{\pi} \left\{ k \cos^{-1}(k) - \sqrt{1-k^2} \right\} \Big|_0^1 = \frac{s}{\pi}\end{aligned}\quad (22)$$

The perimeter of the service area has to be recalculated as $6(m-1)$ one-third arcs and 6 half circles of radius R :

$$6L_{edge} = 6(m-1) \cdot \left(\frac{2\pi R}{3} \right) + 6 \cdot \left(\frac{2\pi R}{2} \right) = 2\pi R(2m+1)\quad (23)$$

For a local service area with n edges bordering another one, the length of borderline eligible to invoke hand-offs is:

$$nL_{edge} = \frac{n\pi R(2m+1)}{3}\quad (24)$$

And we recalculate the service area as well. The area is basically a hexagon with some "decorations" around the perimeter:

$$\begin{aligned}A &= \frac{3\sqrt{3}}{2} \left(\sqrt{3}mR + \frac{R}{\sqrt{3}} \right)^2 \\ &\quad + 6 \left\{ \frac{\pi R^2}{2} - \frac{R^2}{\sqrt{3}} + (m-1) \left(\frac{\pi R^2}{2} - \frac{\sqrt{3}R^2}{4} \right) \right\} \\ &= R^2 \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}\end{aligned}\quad (25)$$

By accumulating the $\delta\bar{P}_{cross}$ along the perimeter and averaging with total area, the probability of a user crosses the borderline for mobile stations located in the service area for $s \ll R$ is:

$$\begin{aligned}\bar{P}_{cross} &= \frac{n\pi R(2m+1)s}{3\pi R^2 \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}} \\ &= \frac{n(2m+1)s}{3R \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}}\end{aligned}\quad (26)$$

Similar to the continuous counterpart, P_{HO} is given by the

following equation:

$$\begin{aligned}P_{HO} &= \frac{P_{cross}}{P_{cross} + P_r} \\ &= \frac{\frac{E(n)(2m+1)s}{3R \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}}}{\frac{E(n)(2m+1)s}{3R \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}} + \frac{\alpha}{\frac{D_{sync}}{BW_s} + T_{ls}}}\end{aligned}\quad (27)$$

for $s \ll R$, where $E(n)$ is the average number of edges bordering another local service area as well, which is 0 and $\frac{24}{7}$ in Configuration A and B, respectively.

3) *Average Response Time Comparison of the Two Configurations:* The average response time of configuration A $T_{response}^A$ is still $2(T_r + T_l^A)$. The average traverse time of Configuration B is equal to its continuous counterpart (12) as well. If we expect that Configuration B would bring a shorter average response time over Configuration A, the upper bound of P_{HO}^B is unchanged:

$$P_{HO}^B < \frac{(\sqrt{7}-1)(4+3\ln(3))}{12\sqrt{3}+2+6\ln(3)} \approx 0.4087356087$$

Therefore, the constraints for m , s , R , $\frac{D_{sync}}{BW_s}$, α , T_{ls} , and T_u are represented in the equation below:

$$\frac{\frac{8(2m+1)s}{R \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}}}{\frac{8(2m+1)s}{R \left\{ \frac{9\sqrt{3}m^2}{2} + \left(2\pi + \frac{3\sqrt{3}}{2} \right) m + \pi \right\}} + \frac{7\alpha}{\frac{D_{sync}}{BW_s} + T_{ls}}} < 0.051091951\quad (28)$$

C. Simulation Result

To verify the estimations of P_{cross} , we use the Monte Carlo method by running a simulation program which sets up base stations of given R at optimal locations, randomly puts a large number of mobile stations, moves them away from their original location a fixed distance in any direction, and measures the number of the mobile stations escaping from the service area.

To compare the errors of the two different approaches, we set two environments with short R and large m , and long R with small m , and adjustable s . In the former environment, we set $R = 0.25$, $m = 40$, s varies from 0.1 to 2.0 with 0.01 steps, and place 10^7 mobile stations. The P_{HO} derived by the estimators and measured in the simulation are compared in Figure 13.

As we can see, the continuous service area approach is a better estimator since the shape of the service area is very close to a perfect regular hexagon in this environment. Furthermore, we compare the error rate of both estimators and compare them in Figure 14.

We can see in this series of simulations, the optimal arranged base stations approach only works well with very low s . However, when we set $R = 2.0$ and $m = 5$ and run the same simulations, it becomes a different story as shown in Figure 15.

Since the base stations are far less dense than in the previous setting, the "ripples" around the service area get larger and distort the shape away from a perfect regular hexagon. As

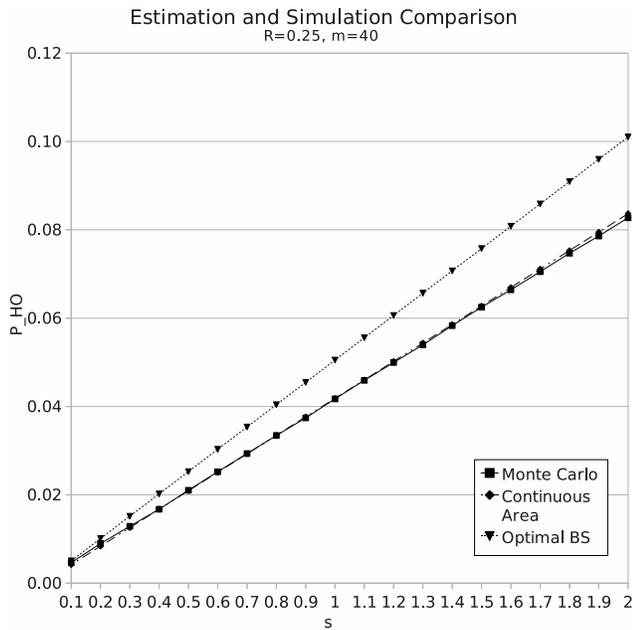


Fig. 13. Comparison of estimations and simulation result with $R = 0.25$ and $m = 40$.

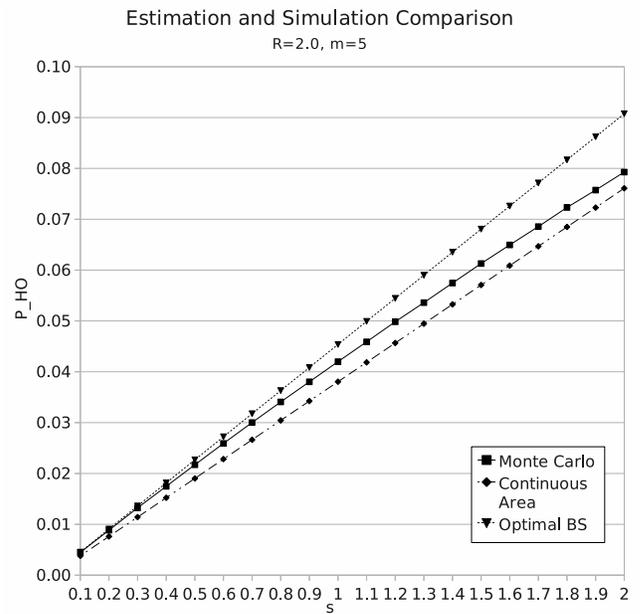


Fig. 15. Comparison of estimations and simulation result with $R = 2.0$ and $m = 5$.

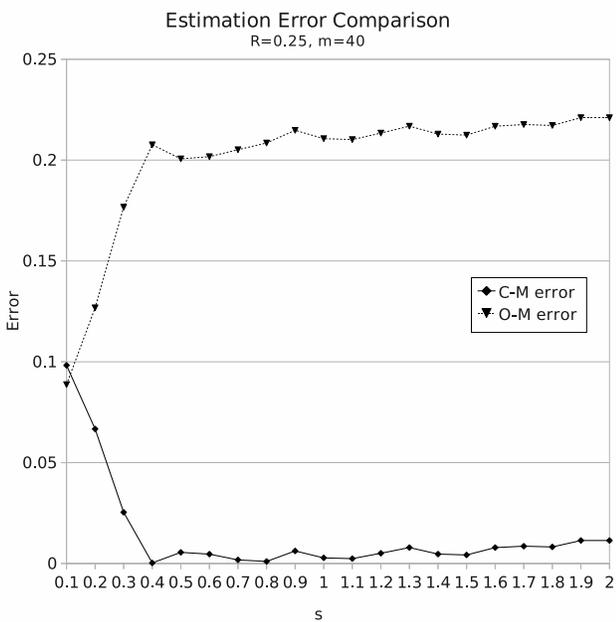


Fig. 14. Comparison of estimation errors with $R = 0.25$ and $m = 40$.

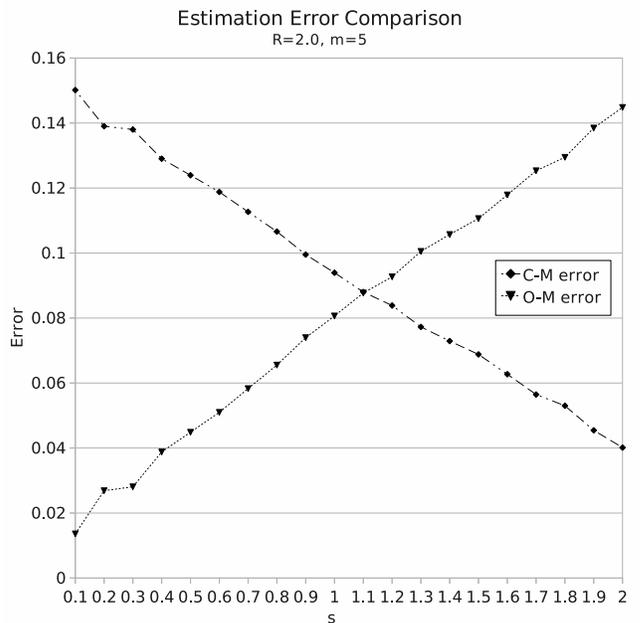


Fig. 16. Comparison of estimation errors with $R = 2.0$ and $m = 5$.

we can see in Figure 15, the optimal arranged base stations approach is a very accurate P_{HO} estimator for $s \leq 0.5$ ($s \leq \frac{R}{4}$), and the continuous service area approach gets more and more accurate P_{HO} in response to increasing user mobility.

By comparing the estimation errors of both approaches in Figure 16, we can see the accuracies of the two estimators significantly depend on user mobility.

VI. PERFORMANCE EVALUATION USING THE UMTS URBAN MOBILITY MODEL

Although the free particle analysis in the previous section relates the overall performance to the MS's mobility and the infrastructure's geographical parameters, this mobility model is too arbitrary to preview the performance of the proposed configuration and hand-off protocol in real world. Since each MS moves without inertia and any intelligent intent, it is able to suddenly turn to an opposite direction in the free particle

mobility model. This moving characteristic only makes sense for an application virtualization service specific for a bunch of drunks wondering on a rural plain. In the proposed distributed service configuration, the free particle model does increase the chance an MS triggers and aborts a hand-off procedure in a short period. Therefore, we need to further evaluate the performance of the proposed configuration and hand-off protocol in more realistic mobility models.

Of course the most realistic usage model comes from the field statistics of a mobile phone carrier. The data is extremely difficult to be obtained for several reasons. For instance, carriers may record the users' moving pattern along with other behaviors and store them in a huge database in general. In most case, they do not do any data mining or organization except for their internal research projects. If an outsider requests a data set about user mobility from a mobile phone carrier, they do not know where the data is even if they are willing to help. Furthermore, those records may involve sensitive user privacy. Mobile phone carriers would reluctant to allow any outsider to get access to the databases to prevent from potential legal issues.

Fortunately, European Telecommunication Standards Institute (ETSI) published a document [15] which described three test environments and user mobility models, which are Indoor Office, Outdoor to Indoor and Pedestrian, and Vehicular ones, as common benchmarks to evaluate potential wireless technologies to develop Universal Mobile Telecommunication System (UMTS). Although the reality of the models is never explicitly justified and Jugl and Boche [16] have extended the mobility model to improve the reality, the original UMTS models still provide a fair reference for mobility related performance evaluation. If more realistic mobility models are available, we can replace the UMTS ones and obtain more accurate configuration parameters.

In this section, we set up a simulation environment referring to the UMTS's Outdoor to Indoor and Pedestrian mobility model, also known as the UMTS urban mobility model, and use the empirical approach to establish the correlations between the performance and the size of each local service area and the capabilities of the network infrastructure. With our proposed modification, we enable the simulation to run for an indefinite period of time without presuming any boundary condition.

A. UMTS Urban Mobility Model

As shown in Figure 17, the UMTS Outdoor to Indoor and Pedestrian test environment is basically a Manhattan-like street structure where MSs move along 30 meters wide streets and are only allowed to change directions with half chance at the intersections, which are 200 meters apart. Each MS's moving speed can be updated every 5 meters with 20% chance, and the new speed is generated by a truncated Gaussian distribution whose mean equals 3 km/h, standard deviation equals 0.3 km/h, and minimum speed equals 0 km/h. All MSs are initially uniformly distributed on the Manhattan-like streets.

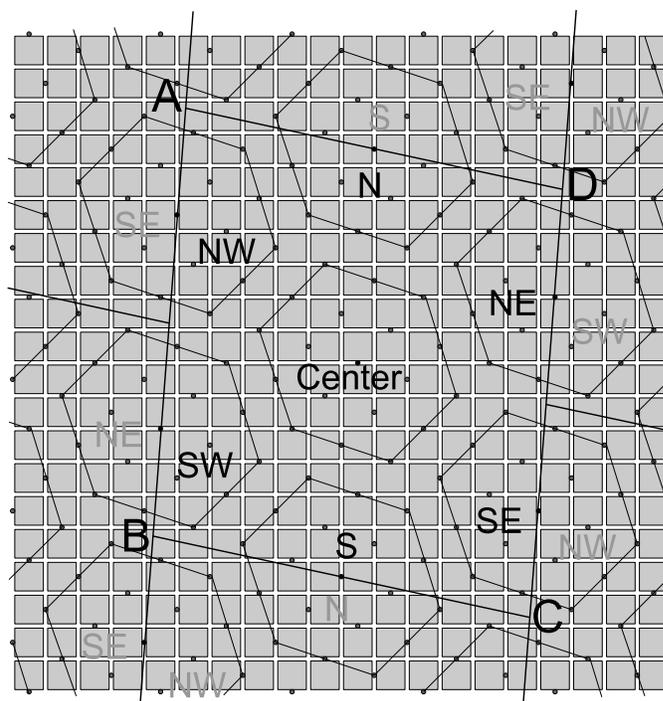


Fig. 17. UMTS outdoor to Indoor and Pedestrian test environment and LSA arrangement.

The UMTS document, however, does not explicitly specify where an MS turns within the intersection area. Therefore, we make a reasonable assumption to overcome the ambiguity. If an MS is supposed to turn in an intersection, it has six points, which are 5 meters apart along the crosswalk, to change its direction before reaching the other side. We assume an MS picks one out of the six points with equal chances as its turning point, which keeps MSs uniformly distributed on the streets rather than be concentrated on a certain part of the streets over time.

The BSs in the UMTS Outdoor to Indoor and Pedestrian test environment are located at the dark grey dots in Figure 17. Although the placement of the BSs is not optimal, it is not far from that. Considering an actual city could be preoccupied by tall private buildings on each block, deploying BSs along the streets makes sense both technically and politically.

One of the shortcomings of the UMTS mobility model is the bounded test area which generates ambiguities on setting boundary conditions. We consequently add some special traffic rules, known as *portals*, to eliminate the boundary discontinuities and allow the interaction among LSAs to be simulated and observed for indefinite period of time. These portals will be described in the next subsection.

B. Möbius City

What interests us is the geographical relation between the service facilities and the MSs' moving space. Once we group the BSs in Figure 17 to form hexagon-shaped LSAs that optimize in both coverage and average transmission distance by deploying servers at the centers, we can find a regular

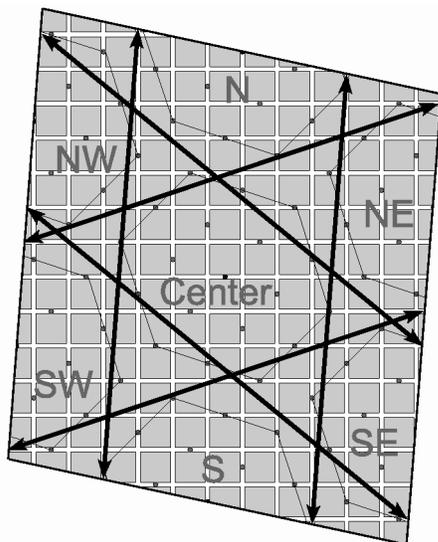


Fig. 18. Möbius City map with teleporting directions.

repetitive pattern of streets and service groups, which depends on N , the number of the BSs per LSA's edge. If we align the origin to a BS, the parallelogram ABCD surrounded by four straight lines, which are:

- 1) $(3N - 1)x + (9N + 5)y = 920(6N^2 + 6N + 2)$ on the north,
- 2) $(3N - 1)x + (9N + 5)y = -920(6N^2 + 6N + 2)$ on the south,
- 3) $(5N + 3)x - (N - 1)y = -920(3N^2 + 3N + 1)$ on the west,
- 4) and $(5N + 3)x - (N - 1)y = 920(3N^2 + 3N + 1)$ on the east,

can be regarded as the element of the repetitive pattern and represent sufficient geographical information we need. We can, therefore, crop out parallelogram ABCD in Figure 17 as our new test area, where we call *Möbius City* as shown in Figure 18, to represent every identical piece comprises the indefinite large test area.

Möbius City only has four LSGs. The center one is the only complete LSA. The north half (N) and the south half (S), the northwest half (NW) and the southeast half (SE), and the northeast half (NE) and the southwest half (SW), comprise the three other LSAs. The latter three LSAs' allocation emulates six complete LSAs around the center one in the original test area. Since we are only interested in when, where, and how frequently an MS moves from one LSA to another rather than specifically identifying which one it moves from and to, assigning only four LSGs is sufficient for our work.

Möbius City is comprised by the area cropped from the original street structure and portals at the boundaries. Just like moving through the tunnels in Pac-Man's maze, whenever an MS moving among the streets reaches a boundary and is about to escape from Möbius City, the portal teleports it to a proper location at the opposite side and reenter Möbius City. The rules of the portals are:

- 1) For MSs about crossing north boundary, teleport them to $(-230(N - 1), -230(5N + 3))$ from their current locations.
- 2) For MSs about crossing south boundary, teleport them to $(230(N - 1), 230(5N + 3))$ from their current locations.
- 3) For MSs about crossing west boundary and their current locations satisfy $3(N - 1)x + (9N + 5)y > 0$, teleport them to $(230(4N + 3), -230(4N + 1))$ from their current locations.
- 4) For MSs about crossing west boundary and their current locations satisfy $3(N - 1)x + (9N + 5)y \leq 0$, teleport them to $(230(5N + 2), 230(N + 2))$ from their current locations.
- 5) For MSs about crossing east boundary, and their current locations satisfy $3(N - 1)x + (9N + 5)y > 0$, teleport them to $(-230(5N + 2), -230(N + 2))$ from their current locations.
- 6) For MSs about crossing east boundary, and their current locations satisfy $3(N - 1)x + (9N + 5)y \leq 0$, teleport them to $(-230(4N + 3), 230(4N + 1))$ from their current locations.

The teleport directions are shown in Figure 18 as well.

An MS moving through a portal doesn't encounter any discontinuity except its coordinates: its direction and speed are the same, it associates with the same LSG, and the geographical parameters relative to the service group's facilities remain. Thus, everything interests us is equivalent as the MS moving into an adjacent parallelogram area in an indefinite large test area.

C. Configuration of Backhaul Network

Although connecting every BS to the corresponding server through a line-of-sight and high-speed direct link offers the lowest transmission latency, constructing such a backhaul network is impractically expensive. Therefore, we assume each BS only has direct connections to its six neighboring BSs to form a mesh network as the core network. In mesh-styled backhaul network, network latency between a BS and the server depends on the number of nodes along the shortest path, the total length of the path, and the relay latency per node. The former two factors are related to the coordinates of the BS and the server, which will be simulated as well.

D. Traverse Delay

We define the response time as the average time interval from a user sends an input till the expected output update is received. The proposed server configuration is meant to improve the response time by reducing traverse delay along the communication route between each BS to the server which is hosting the service. Factors other than the traverse delay, such as computational capabilities provided by servers, would affect the user experience and the quality of our service. Most

of them, however, either affect different configurations equally, or can be overcome with reasonable cost.

Traverse delay is defined as:

$$T_{tv} = 2 \cdot \left\{ \frac{L_r}{V_r} + \frac{L_l}{V_l} + N_{rt} \cdot T_{rt} + N_{rl} \cdot T_{rl} \right\} \quad (29)$$

where L_r is the distance of radio transmission, which is the distance between the MS and the BS it currently uses, V_r is the propagation speed of radio, which equals to the speed of light, L_l is the total length of wireline transmission in the mesh network, V_l is the propagation speed in wireline, which is approximately two thirds of the speed of light, N_{rt} is the number of nodes along the transmission path in the mesh network, T_{rt} is the average waiting time per node in the mesh network, which includes nodal processing delay, queuing delay, and transmission delay, N_{rl} is the number of servers which are receiving the snapshot and relaying data to/from the VM server, and T_{rl} is the processing and relay time per server in the hand-off chain.

E. Hand-off Duration

Whenever a VM-level hand-off occurs, i.e., an MS detects that it's out of the range of the original BS and the nearest BS belongs to another LSG at the latest update, we set up an anticipated hand-off end time by adding hand-off duration to the current time. The hand-off duration is given by the following equation:

$$T_{ho} = T_x + \frac{L_s}{V_l} + N_s \cdot T_{rt} \quad (30)$$

where T_x is the total time to deliver every bit of a snapshot to media, which is the summation of queuing delay, processing delay, and transmission delay of the snapshot, which is proportional to the size of the snapshot, L_s is the total transmission distance between the current and the next VM servers, and N_s is the number of nodes between two neighboring servers, which always equals to $2N + 1$ in this case.

F. Update Time Points and Cost Charging

Updates occur for two reasons: a hand-off is completed, or an MS reaches an update position. At each update time point, T_{tv} and transaction counts are updated concurrently.

Whenever a position update comes at T_{now} , all hand-off end times registered in queue earlier than T_{now} have to be treated as update time points according to the algorithm described below:

- 1) Define T_n as the n^{th} earliest hand-off end time in queue, L_{sn} as the total transmission distance between servers corresponding to the n^{th} earliest hand-off in queue, L_r , L_l , N_{rt} , and N_{rl} are the current cost parameters calculated by the MS's current position and hand-off status, and T_{last} as the previous update time.
- 2) If $T_{now} > T_0$, insert an update time point at T_0 , calculate the transaction counts by the Poisson process given user input rate λ and time duration $(T_0 - T_{last})$, set $T_{last} = T_0$, subtract N_{rl} by one, subtract N_{rt} by

$\{2N + 1\}$, subtract L_l by L_{s0} , update T_{tv} according to the new parameters, and remove T_0 and corresponding L_{s0} from the queues.

- 3) Redo step 2 until $T_{now} < T_0$ or the queue is emptied.
- 4) Calculate the transaction counts by the Poisson process given λ and time duration $(T_{now} - T_{last})$, update T_{tv} according to the new parameters, and set new $T_{last} = T_{now}$.

As specified in UMTS urban mobility model, we update the MSs' positions every 5 meters. Since a hand-off may occur at the same time, we have to handle the extra cost brought by it as well. When a new hand-off occurs with a position update at current time T_{now} while the previous update time is T_{last} , and every hand-off end time earlier than T_{now} is already treated with the above algorithm, we use another algorithm to update cost parameters, which is described below:

- 1) Register the new hand-off end time and the corresponding L_s in the queue.
- 2) Increment N_{rl} by one.
- 3) N_{rt} is recalculated by the MS's current position and added by $\{N_{rl} \cdot (2N + 1)\}$.
- 4) Let L_l equals to the summation of all L_s 's in queue.
- 5) T_{tv} is then updated accordingly.
- 6) The transaction counts are calculated by the Poisson process given λ and time duration $(T_{now} - T_{last})$, and then set new $T_{last} = T_{now}$ for the next update.

Every transaction in an update interval is charged with identical T_{tv} . Note that T_{tv} updated at a time point T is applied to the transactions occur *after* T , while transaction counts calculated at T are placed in the time interval ended at T . Although technically we can create a continuous T_{tv} function and integrate it in each update interval to derive a slightly more accurate T_{tv} , it is unnecessarily complex since T_{tv} variation is negligible within the 5 meters (or less) long path.

G. Traverse Time Accounting

The average T_{tv} per transaction is calculated at the end of 100,000 independent simulations, each lasts 86,400 seconds (one day). The simulation results of variable N , T_{rt} , T_{rl} , T_x , and λ , are presented in the following section.

H. Simulation Results

We first simulate how the size of LSAs affects T_{tv} given *nominal* parameters, which are $T_{rt} = 20ms$, $T_{rl} = 500ms$, $T_x = 600s$, and $\lambda = 1.0$. The simulation result is shown in Figure 19.

As we can see in Figure 19, T_{tv} is high in small LSA configurations due to the higher hand-off occurrence rate. As N increases, T_{tv} first descends, levels for a range of N 's, and then linearly ascends. The descending for low N 's is due to the reduction of hand-off occurrence. The smooth ascending for higher N 's is caused by the higher average number of the nodes along the backhaul route and longer average transmission distance while the hand-off occurrence rate is too low to matter. The flat bottom in between is the result of

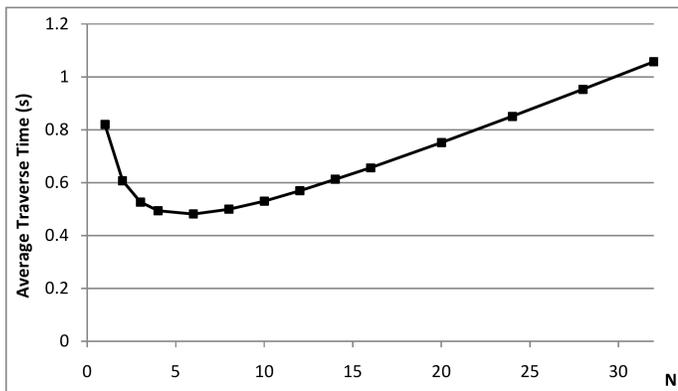


Fig. 19. Simulated T_{tv} of different N given $T_{rl} = 0.5s$, $T_{rt} = 20ms$, and $T_x = 600s$, $\lambda = 1.0$.

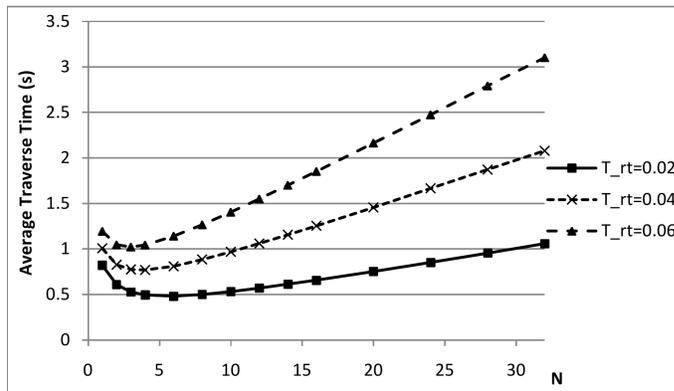


Fig. 21. Simulated T_{tv} given $T_{rt} = 20ms$, $40ms$, $60ms$ and $T_{rl} = 500ms$, $T_x = 600s$, $\lambda = 1.0$.

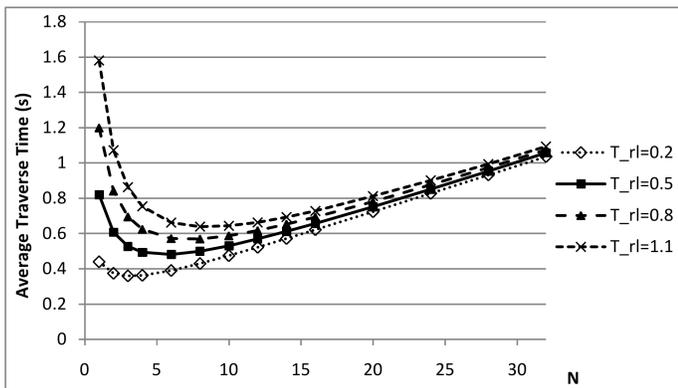


Fig. 20. Simulated T_{tv} given $T_{rl} = 0.2s$, $0.5s$, $0.8s$, $1.1s$ and $T_{rt} = 20ms$, $T_x = 600s$, $\lambda = 1.0$.

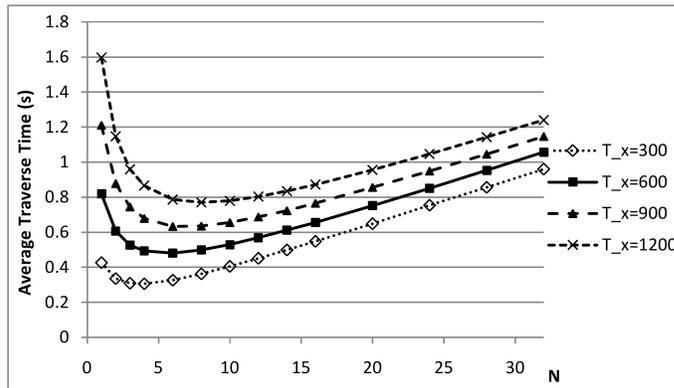


Fig. 22. Simulated T_{tv} given $T_x = 300s$, $600s$, $900s$, $1200s$ and $T_{rt} = 20ms$, $T_{rl} = 0.5s$, $\lambda = 1.0$.

the two effects competing with each other. We can conclude that setting $N = 10$ in this case is optimal in reducing average T_{tv} and keeping the total number of the servers low, which also means lower deployment and maintenance cost.

Since the above conclusion is only applicable in this set of parameters, we adjust each parameter in the nominal set to see how it affects T_{tv} as a function of N in the following subsections.

1) *Effect of T_{rl}* : T_{rl} is the cost that only applies in hand-offs. We set T_{rl} to $200ms$, $800ms$, and $1,100ms$, to see how it affects T_{tv} . The simulated T_{tv} as a function of N and T_{rl} given $T_{rt} = 20ms$, $T_x = 600s$, $\lambda = 1.0$ is shown in Figure 20.

As we can see in Figure 20, higher T_{rl} significantly increases T_{tv} in small LSA configurations. As N increases, T_{tv} given different T_{rl} 's has a tendency to converge together since the hand-off occurrence rate is dramatically reduced and thus renders the effect of T_{rl} insignificant.

2) *Effect of T_{rt}* : Unlike T_{rl} , T_{rt} affects both hand-offs and normal transactions since higher T_{rt} amplifies the influence of transmission distance. The simulated T_{tv} as a function of N and T_{rt} given $T_{rl} = 0.5s$, $T_x = 600s$, $\lambda = 1.0$ are shown in Figure 21.

Figure 21 shows the comparison of T_{tv} 's as functions of N

given $T_{rt} = 20ms$, $40ms$, and $60ms$. We can easily figure out that as T_{rt} increases, not only T_{tv} increases, but it also increases more sharply for higher N and thus compresses the optimal range of N since higher T_{rt} increases the communication cost per transmission distance in the mesh network. In larger LSA configurations, although hand-offs rarely occurs and thus related cost is minimized, the inner-LSA transmission cost increases more significantly due to the higher nodal cost T_{rt} .

3) *Effect of T_x* : T_x affects the cost only in hand-offs. Higher T_x may mean larger synchronization data, longer hand-off initialization time, or longer queuing delay. How T_x affects T_{tv} is represented in Figure 22.

Since T_x is the dominant factor of each hand-off's duration, increasing T_x fairly increases the proportion of the transactions occurred during hand-offs for every N . It is why T_{tv} 's as functions of N given different T_x 's are virtually parallel to each other and show little tendency to converge as N increases.

4) *Effect of λ* : Although not being an intuitive factor, we still simulate T_{tv} 's as functions of N given different user input rates λ . The simulated T_{tv} 's given $\lambda = 0.33$, 0.5 , and 1.0 inputs per second are almost identical. To visualize the differences, the normalized simulation results are compared in Figure 23.

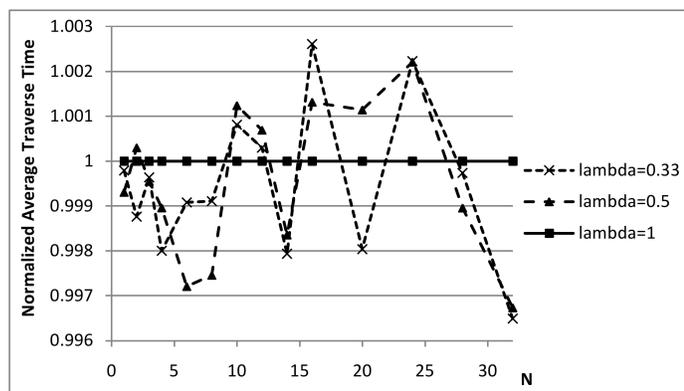


Fig. 23. Normalized simulation results given $\lambda = 0.33, 0.5, 1.0$ and $T_{rt} = 20ms, T_{rl} = 0.5s, T_x = 600s$.

As we can see in Figure 23, there is no difference induces by adjusting λ per se in statistical view. We should keep in mind, however, that the user experience and the maximum tolerable response delay depend on the interactivity of the application software.

VII. PERFORMANCE EVALUATION USING THE UMTS RURAL MOBILITY MODEL

In the previous section, we used UMTS's urban mobility model to empirically establish the correlations between the performance and the size of each local service area and the capabilities of the network infrastructure. In this section, we employ the Vehicular test environment, also known as the rural vehicular mobility model, of the UMTS document [15] to complete the performance evaluation of the proposed configuration and protocol. The other test environment, that is, the Indoor Office environment described in the UMTS document, will not be discussed in this paper since the communication distance varies relatively small. The Indoor Office environment is more relevant to the radio and baseband design, which is out of our scope. In the UMTS rural vehicular mobility model, BSs are sparsely but optimally placed, MSs move faster and more freely, and the hand-off behavior among base stations is different as well. Although the simulation program in the UMTS rural vehicular model is significantly different from the one presented in the previous section, the concept of the indefinite simulation is retained.

A. UMTS Vehicular Mobility Model

As shown in Figure 24, the UMTS rural vehicular test environment is a plain with no physical obstacle. Each MS's speed is fixed at 120 km/h. Each MS's moving direction is allowed to change up to 45° left or right every 20 meters with 20% chance. All MSs are initially uniformly distributed on the plain.

The BSs in the UMTS rural vehicular test environment are located at the dark grey dots in Figure 24. Each BS has three directional antennae to serve tri-sector cells. Each cell is assumed to be a hexagon and seamlessly tiles with each other. Each cell's radius R is either 2,000 meters (for services up

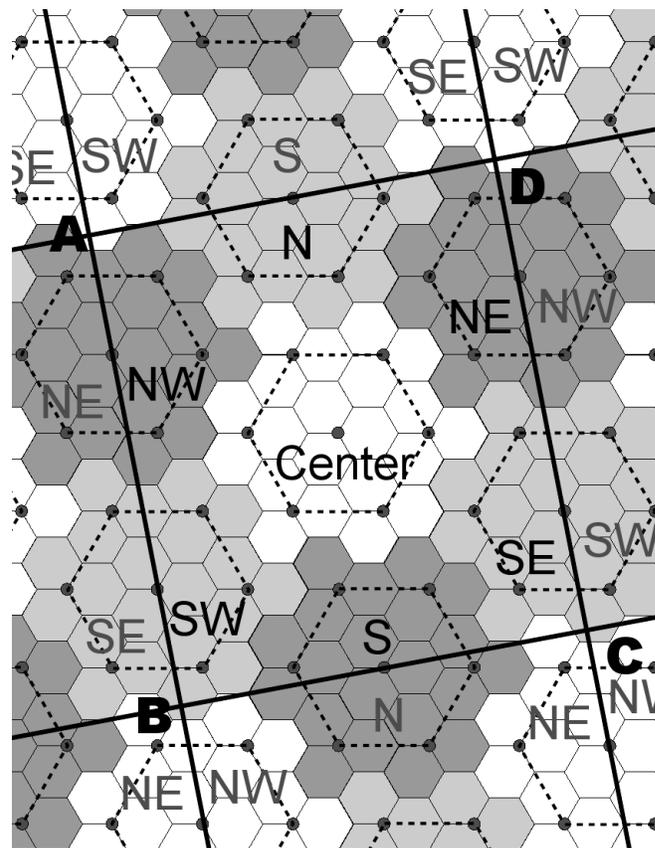


Fig. 24. The UMTS rural vehicular test environment with LSA arrangement.

to 144kbit/s) or 500 meters (for services above 144kbit/s). Therefore, the minimum distance between two BSs can be 6 km or 1.5 km, respectively.

The original UMTS mobility model generates discontinuities on the boundaries of the test area. We consequently add some special traffic rules, known as *portals*, to eliminate the boundary discontinuities and allow the interaction among LSAs to be simulated and observed for an indefinite period of time. The characteristics of the portals will be detailed in the next section.

B. Möbius County

What interests us is the geographical relation between the service facilities and the MSs' moving space. As the method we conducted in the previous section, the first step is to define a sample area which can represent all the geographical characteristics of service infrastructure we need. We first group the BSs in Figure 24 to form approximately hexagon-shaped LSAs which are optimized in both coverage and average transmission distance by deploying servers at the centers. As the urban counterpart, i.e., Möbius City, in the previous section, the sample area should include one complete LSA in the center and six neighboring halves. Given R and N , the number of the BS intervals per LSA's edge, if we align the origin to the server of an LSG, we define the Parallelogram ABCD surrounded by four straight lines, which are:

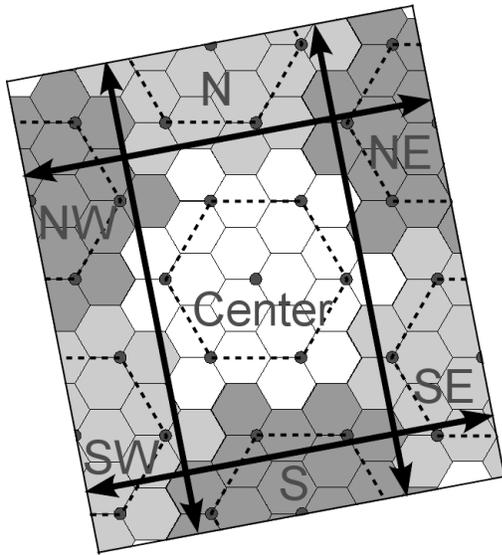


Fig. 25. Möbius County map with teleporting directions.

- 1) $\sqrt{3}x - 3(2N + 1)y = -6\sqrt{3}R(3N^2 + 3N + 1)$ on the north,
- 2) $\sqrt{3}x - 3(2N + 1)y = 6\sqrt{3}R(3N^2 + 3N + 1)$ on the south,
- 3) $\sqrt{3}(2N + 1)x + y = -3\sqrt{3}R(3N^2 + 3N + 1)$ on the west,
- 4) and $\sqrt{3}(2N + 1)x + y = 3\sqrt{3}R(3N^2 + 3N + 1)$ on the east.

as the sample area of our best interest. We can, therefore, crop out Parallelogram ABCD in Figure 24 as our test area, where we call *Möbius County* as shown in Figure 25, to represent every identical piece comprises the indefinite large test area.

Like Möbius City, assigning four logical LSGs in Möbius County is sufficient to figure out when, where, and how frequently an MS moves from one LSA to another. However, to apply the hand-off aborting mechanism, which was disabled in the previous section, we need to distinguish whether an MS is coming back to the LSA it just left or entering the LSA on the opposite side of the one it just crossed. Therefore, we have to assign an additional unique identification for each LSG.

The portals around Möbius County are also similar to those around Möbius City. Whenever an MS is about escaping from Möbius County, the portal teleports it to a proper location at the opposite side so that it reenters Möbius County. Therefore, Möbius County can emulate a limitless test area. Since there is no street structure to align in Möbius County, the rules of the portals are much more simple and straightforward than of Möbius City:

- 1) For MSs about crossing the north boundary, teleport them to $(3R, -3\sqrt{3}R(2N + 1))$ from their current locations.
- 2) For MSs about crossing the south boundary, teleport them to $(-3R, 3\sqrt{3}R(2N + 1))$ from their current locations.

- 3) For MSs about crossing the west boundary, teleport them to $(-\frac{9R(2N+1)}{2}, -\frac{3\sqrt{3}R}{2})$ from their current locations.
- 4) For MSs about crossing the east boundary, teleport them to $(\frac{9R(2N+1)}{2}, \frac{3\sqrt{3}R}{2})$ from their current locations.

The teleport directions are shown in Figure 25 as well.

The purpose of the portals is to eliminate all discontinuities except the MS's coordinates when it is moving out of the boundary: it keeps the same direction and speed, it associates with the same logical LSG, and preserves the geographical parameters relative to the service group's facilities. Thus, everything interests us is equivalent as the MS moving into an adjacent parallelogram area in a limitless test area.

C. Configuration of Backhaul Network

We assume a mesh-styled backhaul network as we did in the previous section. Therefore, each BS only has direct links to its six neighboring BSs. In the mesh-styled backhaul network, network latency between a BS and the server depends on the number of nodes along the shortest path, the total length of the path, and the relay latency per node. The former two factors are related to the coordinates of the BS and the server, while the last one is varied to simulate different nodal transmission capabilities.

D. Performance Metric and Hand-off Duration

The definition of the traverse delay is identical to the counterpart in the previous section:

$$T_{tv} = 2 \cdot \left\{ \frac{L_r}{V_r} + \frac{L_l}{V_l} + N_{rt} \cdot T_{rt} + N_{rl} \cdot T_{rl} \right\}$$

The hand-off duration is also the same as in the previous section:

$$T_{ho} = T_x + \frac{L_s}{V_l} + N_s \cdot T_{rt}$$

E. Update Time Points and Cost Charging

This part of our simulation program is virtually identical to the counterpart in the previous section. The only differences are: 1) the position update interval is 20 meters instead of 5 meters, 2) the time increment is fixed at 0.6 seconds since each MS's moving speed is always 120 km/h.

F. Traverse Time Accounting

The average T_{tv} per transaction is calculated at the end of 100,000 independent simulations, each lasting 86,400 seconds. The simulation results of variable N , T_{rt} , T_{rl} , T_x , and λ for both $R = 2,000m$ or $500m$, are presented in the following section.

G. Simulation Results

We first simulate how the size of LSAs affects T_{tv} given nominal parameters, which are $T_{rt} = 20ms$, $T_{rl} = 500ms$, $T_x = 600s$, and $\lambda = 1.0$. The simulation results of both R settings are shown in Figure 26.

As we can see in Figure 26, both T_{tv} 's bear a strong resemblance in shape to the counterpart in the previous section despite the significantly different mobility models. T_{tv} 's are

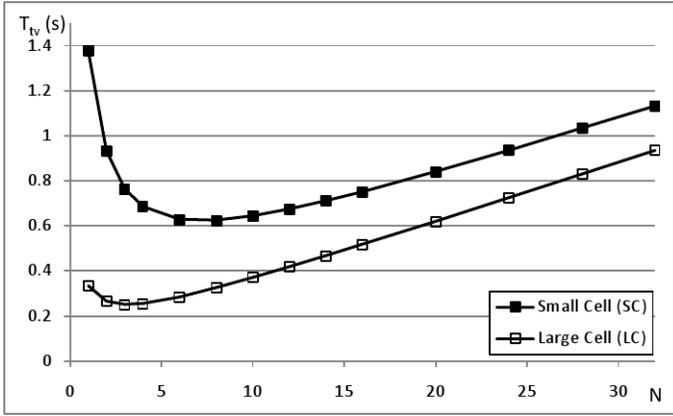


Fig. 26. Simulated T_{tv} of different N of both cell configurations given $T_{rl} = 0.5s$, $T_{rt} = 20ms$, and $T_x = 600s$, $\lambda = 1.0$.

high in small LSA configurations due to the higher hand-off occurrence rate. As N increases, T_{tv} 's first descend, level for several N 's, and then linearly ascend. The descending for low N 's is due to the reduction of hand-off occurrences. The smooth ascending for higher N 's is caused by the higher average number of the nodes along the backhaul route and the longer average transmission distance while the hand-off occurrence rate is too low to matter. The flat bottom in between is the result of the two effects competing with each other.

Note although we compare two cell configurations, $R = 2,000m$ and $R = 500m$, in the same figure, each LSA of the former one is in fact 4 times larger than of the latter one. Therefore, each MS encounters much fewer hand-offs in the large cell configuration than in the small cell one. We can also observe slightly steeper ascending for higher N 's in the large cell configuration than in the small cell one due to the higher propagation delay brought by the longer wireline and wireless transmission distances.

We can conclude that in this case, setting $N = 4$ for the large cell configuration, and $N = 8$ for the small cell one, are optimal in reducing average T_{tv} and keeping the total number of the servers low, which also means lower deployment and maintenance cost.

Since the above quantitative conclusion is only applicable in this set of parameters, we adjust each parameter in the nominal set and compare the results to see how it affects T_{tv} 's as functions of N in the following subsections.

1) *Effect of T_{rl}* : T_{rl} only participates in hand-off conditions. In this simulation, we set T_{rl} to $200ms$, $800ms$, and $1,100ms$, and see how it affects both T_{tv} 's. Both simulated T_{tv} 's in large and small cell configurations as functions of N and T_{rl} given $T_{rt} = 20ms$, $T_x = 600s$, $\lambda = 1.0$ are shown in Figure 27.

As we can see in Figure 27, higher T_{rl} significantly increases T_{tv} 's in small LSA configurations due to the higher occurrence rate of hand-offs. As N increases, T_{tv} 's in each cell configuration given different T_{rl} 's have a tendency to converge together since the hand-off occurrence rate is dramatically reduced and thus renders the effect of T_{rl} insignificant. In

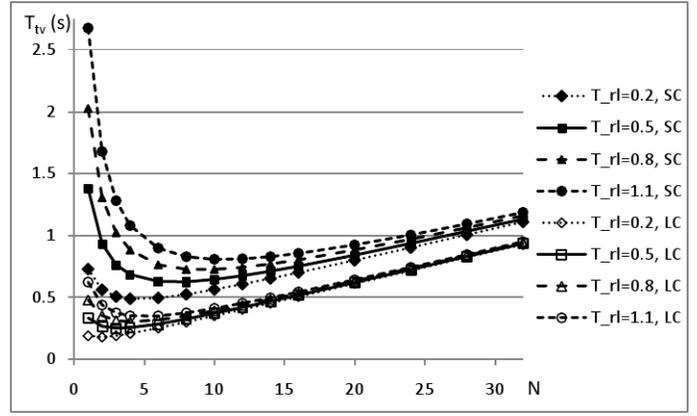


Fig. 27. Simulated T_{tv} 's of both cell configurations given $T_{rl} = 0.2s, 0.5s, 0.8s, 1.1s$ and $T_{rt} = 20ms$, $T_x = 600s$, $\lambda = 1.0$.

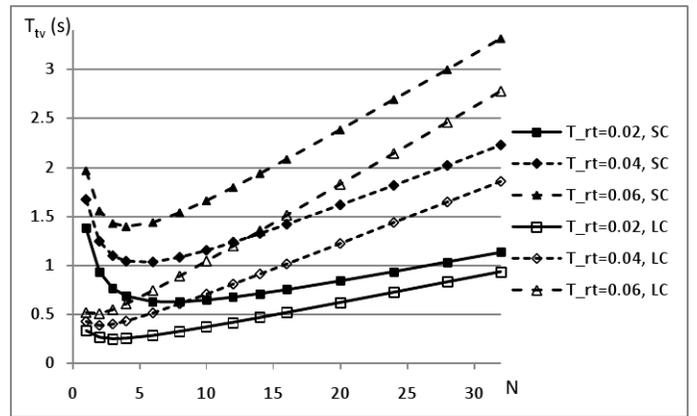


Fig. 28. Simulated T_{tv} 's of both cell configurations given $T_{rt} = 20ms, 40ms, 60ms$ and $T_{rl} = 500ms$, $T_x = 600s$, $\lambda = 1.0$.

the large cell configuration, T_{tv} 's converge more significantly and earlier due to the extremely low hand-off occurrence rate.

2) *Effect of T_{rt}* : Higher T_{rt} amplifies the influence of transmission distance. The simulated T_{tv} 's in both cell configurations as functions of N and T_{rt} given $T_{rl} = 0.5s$, $T_x = 600s$, $\lambda = 1.0$ are shown in Figure 28.

Figure 28 shows the comparison of T_{tv} 's of both cell configurations as functions of N given $T_{rt} = 20ms, 40ms$, and $60ms$. Besides the resemblance in shape to the counterpart in the previous section, we can also notice that T_{rt} is a more decisive factor for the large cell configuration's performance due to the low hand-off occurrence rate and the long average communication distance in each LSA. Even $N = 1$ can be preferable if T_{rt} is greater than $60ms$ in the large cell configuration.

3) *Effect of T_x* : T_x only affects the cost brought by hand-offs. A higher T_x may mean a larger snapshot file, a longer hand-off initialization time, or a longer queuing delay. How T_x affects T_{tv} is represented in Figure 29.

Similar to the counterpart in the previous section, T_{tv} 's of each cell configuration as functions of N given different T_x 's are virtually parallel for high N to each other and show very

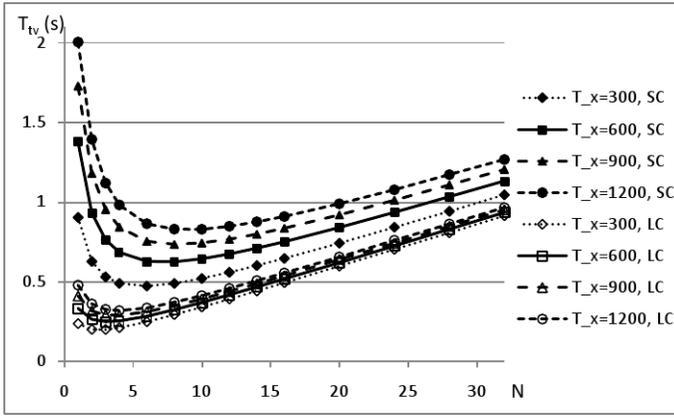


Fig. 29. Simulated T_{tv} of both cell configurations given $T_x = 300s, 600s, 900s, 1,200s$ and $T_{rt} = 20ms, T_{rl} = 0.5s, \lambda = 1.0$.

little tendency to converge as N increases. However, slightly higher optimal N brought by higher T_x in both configurations is still observable.

4) *Effect of λ* : Although we have shown that user input rate λ was not a relevant parameter in the previous section, we still simulate T_{tv} 's as functions of N given different user input rates λ in Möbius County. We again confirm that the property doesn't change in the UMTS rural vehicular mobility model.

However, we should keep in mind that the user experience depends more on the interactivity of the application software than on the absolute response latency.

VIII. CONCLUSION

In this paper, we have first proposed a geographically distributed server arrangement and a hand-off protocol for application virtualization services for mobile users. We have also proposed two analyses to evaluate the impact and the benefit of utilizing the proposed hand-off protocol. And we verify the estimators of probability of a user crossing the borderline by the Monte Carlo experiments and evaluate the accuracies and limitations of both approaches.

After going through the quantitative approaches to compare different server-user configurations, we find out the factors which should be taken into consideration when a service provider plans to launch virtual application services or even virtual desktop services on mobile devices. If they analyze the user behaviors and the application's runtime properties and conclude that their users rarely move, or only move at a low speed, or the data volume required to recreate the runtime environment is relatively small, it is more likely to improve the performance by geographically deploying more servers to cover the whole service area and implement the proposed hand-off protocol. On the other hand, should one or more factors induce a very high hand-off count or overhead, the conventional single server configuration would be preferred.

Following the quantitative approaches, we proposed Möbius City and Möbius County, which are based on the original UMTS urban and rural mobility models but modified to enable

MSs to move in the test environment for indefinite period of time without presuming any boundary condition. We simulate the network delay as a result of MSs movements and the occurrences of VM-level hand-offs in Möbius City and Möbius County given variable sizes of LSAs, server relay latencies, routing costs, and transmission delays of snapshots.

By using Möbius City and Möbius County as the test environments, we can evaluate the performance impact and benefit of different sizes of LSAs and infrastructure technologies and capabilities before providing an application virtualization service for mobile computing devices. Möbius City and Möbius County simulations can provide performance previews for planning network infrastructures aim to improve application virtualization services on unknown urban and rural areas, respectively.

APPENDIX A

The detail derivation of (6):

$$\begin{aligned}
 \bar{P}_{cross}(L, s, n) &= \frac{2}{3\sqrt{3}L^2} \int_0^s \{n(L-2s) \cdot P_{cross}(d, s)\} dd \\
 &+ ns^2 \left(2 - \frac{1}{\sqrt{3}}\right) \cdot \frac{2\bar{P}_{cross}}{3\sqrt{3}L^2} \\
 &= \frac{2}{3\pi\sqrt{3}L^2} \int_0^s \left\{n(L-2s) \cdot \cos^{-1}\left(\frac{d}{s}\right)\right\} dd \\
 &+ \frac{2ns^2(2\sqrt{3}-1) \cdot \bar{P}_{cross}}{9L^2} \\
 &= \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} \int_0^1 \cos^{-1}(k) dk \\
 &+ \frac{2ns^2(2\sqrt{3}-1) \cdot \bar{P}_{cross}}{9L^2} \\
 &= \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} \cdot \left\{k \cos^{-1}(k) - \sqrt{1-k^2}\right\} \Big|_0^1 \\
 &+ \frac{2ns^2(2\sqrt{3}-1) \cdot \bar{P}_{cross}}{9L^2} \\
 &= \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} + \frac{2ns^2(2\sqrt{3}-1) \cdot \bar{P}_{cross}}{9L^2}
 \end{aligned}$$

APPENDIX B

We assume an MS is at the edge of a service area during hand-offs. Due to the symmetry of hexagons, the average distance from an arbitrary point at the edge of a hexagon to its center is equivalent to the average distance from the 30-degree vertex to an arbitrary point along the opposite leg as shown

in Figure 4.

$$\begin{aligned} & \int_0^{\frac{L}{2}} \left\{ \sqrt{\frac{3L^2}{4} + h^2} \right\} dh \\ &= \left\{ \frac{h}{2} \sqrt{\frac{3L^2}{4} + h^2} + \frac{3L^2}{8} \ln \left| h + \sqrt{\frac{3L^2}{4} + h^2} \right| \right\} \Big|_0^{\frac{L}{2}} \\ &= \frac{L^2}{4} + \frac{3L^2 \ln(3)}{16} \end{aligned}$$

By averaging the result above along the leg, the average transmission distance for an MS at the edge of a service area is:

$$\frac{\frac{L^2}{4} + \frac{3L^2 \ln(3)}{16}}{\frac{L}{2}} = \frac{L}{2} + \frac{3L \ln(3)}{8}$$

Since propagation delay is proportional to the transmission distance, T_{lmax} can be presented in terms of T_{ls} and L :

$$\begin{aligned} \frac{T_{lmax}}{T_{ls}} &= \frac{\frac{L}{2} + \frac{3L \ln(3)}{8}}{\sqrt{3}L} \\ T_{lmax} &= \left\{ \frac{1}{2\sqrt{3}} + \frac{3 \ln(3)}{8\sqrt{3}} \right\} T_{ls} \end{aligned}$$

APPENDIX C

The detail derivation of (16):

$$\begin{aligned} & \frac{T_l}{\sqrt{12}} \left\{ 1 + P_{HO}^C \left\{ \frac{12\sqrt{3} + 2 + 6 \ln(3)}{4 + 3 \ln(3)} \right\} \right\} \\ & < \frac{T_l}{\sqrt{7}} \left\{ 1 + P_{HO}^B \left\{ \frac{12\sqrt{3} + 2 + 6 \ln(3)}{4 + 3 \ln(3)} \right\} \right\} \\ & \sqrt{7} \{1 + P_{HO}^C \cdot k\} < \sqrt{12} \{1 + P_{HO}^B \cdot k\} \\ & k \{ \sqrt{7} P_{HO}^C - \sqrt{12} P_{HO}^B \} < \sqrt{12} - \sqrt{7} \\ & k \left\{ \sqrt{7} \cdot \frac{2\sqrt{3} E_C(n) \left(\frac{\sqrt{12}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^C \right\}}{2\sqrt{3} E_C(n) \left(\frac{\sqrt{12}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^C \right\} + 9\pi\alpha_C} \right. \\ & \quad \left. - \sqrt{12} \cdot \frac{2\sqrt{3} E_B(n) \left(\frac{\sqrt{7}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^B \right\}}{2\sqrt{3} E_B(n) \left(\frac{\sqrt{7}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^B \right\} + 9\pi\alpha_B} \right\} \\ & < \sqrt{12} - \sqrt{7} \\ & \frac{8\sqrt{21} \left(\frac{\sqrt{12}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + \sqrt{\frac{7}{12}} T_{ls}^B \right\}}{8\sqrt{3} \left(\frac{\sqrt{12}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + \sqrt{\frac{7}{12}} T_{ls}^B \right\} + 9\pi\alpha_C} \\ & \quad - \frac{\frac{96}{7} \left(\frac{\sqrt{7}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^B \right\}}{\frac{16\sqrt{3}}{7} \left(\frac{\sqrt{7}s}{L} \right) \cdot \left\{ \frac{D_{sync}}{BW_s} + T_{ls}^B \right\} + 3\pi\alpha_B} \\ & < \frac{(\sqrt{12} - \sqrt{7})(4 + 3 \ln(3))}{12\sqrt{3} + 2 + 6 \ln(3)} \end{aligned}$$

where

$$k = \frac{12\sqrt{3} + 2 + 6 \ln(3)}{4 + 3 \ln(3)}$$

REFERENCES

- [1] Joeng Kim, Ricardo A. Baratto, and Jason Nieh, An Application Streaming Service for Mobile Handheld Devices, *SCC'06 IEEE International Conference on Services Computing*, Sept. 2006, pp. 323-326.
- [2] VMware Inc., "VMware ThinApp Agentless Application Virtualization Overview."
- [3] Ana Fernandez Vilas et al., Providing Web Services over DVB-H: Mobile Web Services, *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 2, May 2007, pp. 644-652.
- [4] Apple Inc., "App Store Review Guidelines for iOS apps," 2.7 and 2.8, <http://developer.apple.com/appstore/guidelines.html>, Retrieved 9 Sep. 2010.
- [5] Google Inc., "Android Market Developer Distribution Agreement," 4.5, <http://www.android.com/us/developer-distribution-agreement.html>, Retrieved 22 Feb. 2011.
- [6] VMware Inc., "VMware MVP (Mobile Virtualization Platform)," <http://www.vmware.com/products/mobile/overview.html>, Retrieved 7 Aug. 2011.
- [7] Chung-Ping Hung and Paul S. Min, Infrastructure Arrangement for Application Virtualization Services, *The 9th International Information and Telecommunication Technologies Symposium (I2TS 2010)*, 2010, pp. 78-85.
- [8] Chung-Ping Hung and Paul S. Min, Service area optimization for application virtualization using UMTS mobility model, *International Conference on Internet Computing (ICOMP 2011)*, 2011, pp. 128-134.
- [9] Chung-Ping Hung and Paul S. Min, Performance evaluation of distributed application virtualization services using the UMTS mobility model, *The First International Conference on Mobile Services, Resources, and Users (MOBILITY 2011)*, 2011, pp. 83-89.
- [10] Marcin Bienkowski et al., Competitive Analysis for Service Migration in VNets, in *Proc. 2nd ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, 2010, pp. 17-24.
- [11] Dushyant Arora, Anja Feldmann, Gregor Schaffrath, and Stefan Schmid, On the benefit of virtualization: strategies for flexible server allocation, *Hot-ICE'11 Proceedings of the 11th USENIX conference on Hot topics in management of internet, cloud, and enterprise networks and services*, 2011.
- [12] L. Peter Deutsch and B. W. Lampson, *SDS 930 Time-sharing System Preliminary Reference Manual*, Doc. 30.10.10, Project Genie, Univ. Cal. at Berkeley, April 1965.
- [13] VMware Inc., "Virtual Desktop Infrastructure."
- [14] R. Buckminster Fuller, *Synergetics: explorations in the geometry of thinking*, Macmillan Publishing Company, 1975.
- [15] ETSI. Universal Mobile Telecommunications System (UMTS); selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03, version 3.2.0), Technical report, European Telecommunication Standards Institute, Apr. 1998.
- [16] H. Boche and E. Jugl, Extension of ETSI's Mobility Models for UMTS in Order to Get More Realistic Results, *Proc. UMTS Workshop*, Günzburg, Germany, Nov. 1998.

Vision towards an Open Electronic Wallet on NFC Smartphones

Glenn Ergeerts*, Dries Schellekens[†], Frederik Schrooyen*, Rud Beyers*, Kevin De Kock*, and Thierry Van Herck*

**Department of Applied Engineering
Artesis University College Antwerp
Antwerp, Belgium*

Email: glenn.ergeerts@artesis.be

*†Department of Electrical Engineering, ESAT/SCD-COSIC and iMinds
KU Leuven
Leuven, Belgium*

Email: dries.schellekens@esat.kuleuven.be

Abstract—Many recent initiatives indicate an evolution towards an electronic wallet to perform all sorts of electronic transactions, such as for example micro payments, loyalty, and transport ticketing. Furthermore, the smartphone is emerging as an indispensable tool for many, containing more and more personal information. Hence, it seems like the ideal medium for carrying the electronic wallet as well. The fast and intuitive touch-and-go philosophy and the integration in mobile devices, makes Near Field Communication (NFC) the perfect technology for contactless electronic transactions. However, the complex ecosystem is holding back the world-wide integration of this technology in mobile handsets, resulting in a low market penetration of NFC smartphones. In our work, we investigated how an NFC-enabled smartphone can be used as an electronic wallet in an existing solution that originally relied on DESFire tags. We implemented a Java Card applet that is compatible with the DESFire specification and explored how this applet can be deployed on the different possible solutions for the Secure Element (SE), such as an active Bluetooth sticker, a SIM card, and a secure microSD card. Finally, the user-wallet interface was examined carefully, more specifically in terms of accessibility of the wallet content for the different smartphone platforms. Based on the overall experiences and results the final conclusions upon the vision towards an open electronic wallet of the future were drawn.

Keywords—NFC; electronic wallet; secure element; SCWS; Java Card; DESFire; smartphone.

I. INTRODUCTION

Today a lot of multinationals including Google, Apple, and Microsoft are involved in an electronic wallet project one way or another. Because of the difficult ecosystem, everything is kept rather closed, which is not very stimulating for the penetration and use of it. This paper offers a more open solution and gives an overview of what is possible and what is not.

Implementing an electronic wallet on smartphones offers a lot of advantages compared to a classic wallet. For example, it offers a solution for containing all today's electronic payment cards; so in case of a payment transaction a customer only needs to select the appropriate payment method. It saves a lot of time and it can be used to hold much more, for instance loyalty cards, drink/entrance vouchers and coupons.

In the case of a festival or an event, visitors are required to obtain drink and food vouchers before they can make an order. This makes them lose precious time by standing in line to exchange their money for vouchers. With the aid of their smartphone, they are capable of carrying out over-the-air transactions and consultations. So essentially, we could throw away our wallet and replace it with our personal smartphone, which we carry already with us all the time now [1].

Mobile and contactless payment systems are finally getting the push in the back they deserve through the upcoming of NFC [2] in handsets, a contactless communication technology designed for electronic payments. Several big companies and research institutes stated that in 2011 up to 50 million NFC handsets were shipped and about 20% of the Points of Sale that were sold had NFC support [3]. These numbers are expected to emerge to 800-1000 million NFC-enabled handsets and up to 80% for Points of Sale with contactless support. Most mobile phones will be NFC-enabled through the use of an integrated NFC controller while others will use a more intermediate solution like a NFC-enabled microSD or active NFC sticker; this paper will discuss all these solutions in more detail. The total market size for mobile payments with NFC is targeted at 680 billion dollar in 2016 [3].

The electronic wallet, that is being developed and tested in the EVENT project funded by the Flemish government, covers a broad range of solutions for the user, from NFC tags to NFC-enabled smartphones. The implemented solution is meant to be open, generic and hybrid. The openness refers to the fact that the wallet will be able to hold several types of payment solutions: existing initiatives such as PingPing (Belgacom) and PayWave (VISA) can be plugged into the wallet, at least from a technical point of view. The design of the wallet will be generic, allowing for usage in a number of different settings. Examples include payments (large as well as small amounts), ticketing, coupons, etc. The hybrid aspect is referring to the fact that value can be stored on the card (offline) as well as on a server (online). Both approaches

have their advantages, but this paper will mainly focus on the offline wallet. In the online use case, the wallet is only containing some sort of identifier for authentication and the user interface afterwards, is more or less comparable with the online banking solutions today. Unlike in the offline case, there is no direct link needed between the SE and the OS of the users phone. This will simplify the user interface issue for consulting the wallets content to a plain web interface, though it makes the payment infrastructure completely reliable on a network connection. Although this will be perfectly acceptable in fixed environment, most of the big event organizers, either professional or leisure, will not want to take the risk.

The security issue is also addressed briefly. The wallet has to be deployed on a SE. Since payment is a critical application, a secure solution must be provided and also the consumer needs to be convinced that the system is at least as safe as the traditional payment methods used today, meaning there is no room for tampering with the data that represents money or tickets that were bought. The possible choices of will be discussed more thoroughly later in this paper.

As mentioned earlier the wallet itself can reside both on a passive DESFire tag or an NFC-enabled device and it relies upon terminals equipped with NFC technology to initiate the actual transactions. The latter holds a number of intrinsic advantages over tags such as allowing users to view and interact with the contents of the wallet, whereas tags rely solely on terminals instead, which can be deemed to be a shortcoming. Since current terminals are intended to be compatible with only passive DESFire tags, one of the goals is that the mobile phone counterpart adopts the current protocols used between terminals and tags. A DESFire applet will be implemented and deployed on a Java Card runtime environment to ensure backwards compatibility with the currently used system.

Finally, there is a lot of differentiation between the various handsets currently available on the market. This, in combination with the difficult accessibility of the Secure element, means that the graphical representation of the content of the wallet is quite a challenge. This paper can be divided in two large sections: Section II will highlight the interface between the wallet and the terminal, whereas Section III focusses on the graphical user interface. Finally, in Section IV we will draw some conclusions.

II. TERMINAL INTERFACE

In this section, we will describe the interface between the terminal and the wallet. In Section II-A we will describe how to achieve backwards compatibility with existing tags. Next in Section II-B, we will discuss the development of a DESFire applet. Finally, we will give an overview of the different deployment options in Section II-C.

A. Terminal interface compatibility with passive tags

In earlier work [4], we implemented an electronic ticketing solution based on MIFARE Classic tags. Even though most secure elements of NFC-enabled smartphones emulate a MIFARE tag, we decided not use this technology in the EVENT project, primarily because the security of MIFARE Classic has been seriously compromised [5], [6], [7].

We selected the MIFARE DESFire EV1 product family from NXP Semiconductors instead, because it offers a flexible file system and strong cryptographic mutual authentication. DESFire tags are popular for public transportation, staff or student identification, building access control and canteen payments, and they are available with a non-volatile memory size of 2 to 8 KB. We believe that this technology is well suited for the realization of an open hybrid electronic wallet.

DESFire tags support up to 28 applications and every application can store up to 32 files. Consequently the same physical tag can be used for multiple applications, such as professional or leisure events, public transport, loyalty programs for supermarkets, etc. Files will typically be used to store an identifier (e.g., to identify a customer in a loyalty program or to link to an online banking account), an offline stored counter that represents an amount of vouchers, or a transaction log.

Another attractive feature of the DESFire standard is its fine-grained access control mechanism: up to 14 different keys can be associated with an application and access rights to the files within the application can be enforced based on these keys. Within the EVENT project we implemented drink/food vouchers as DESFire value files and use separate keys to authorize the credit and debit operation on these files. Terminals at the booths where vouchers can be purchased, will authenticate to the tag using the key that authorizes the credit operation, whereas terminals at a bar or a food stand, will use the key that grants the debit right to redeem a voucher. In the project we decided to make the credit key unique for every tag, by using the tags UID (unique identifier) in a key diversification function, but to keep the debit key fixed for the event and the same for all tags. This design choice was made because of a trade-off between transaction speed, scalability and security. On the one hand debit transactions are fast and scalable as terminals will always use the same key. On the other hand credit transactions are a bit slower as terminals must derive the diversified, tag specific key at the start of the transaction, but they offer a high level of security. As for the consult key we decided to leave the security open, which implicates that anybody holding an NFC mobile device can perform a consultation operation. In our opinion this might stimulate the demand for an NFC-enabled smartphone, since the early adopters will demonstrate the possibilities to their peers. If desired, or in case harm is caused, you can easily protect

this operation as well of course.

As motivated in the introduction, it is crucial that our smartphone implementation of the electronic wallet provides an interface to the terminals that is fully compatible with the regular tags. The DESFire standard supports the wrapping of native commands in ISO7816 APDUs (Application Protocol Data Units), which makes it possible to implement the DESFire protocol on any contactless smart card. The terminal software that we developed in the EVENT project uses these wrapped ISO7816 APDUs, instead of native DESFire APDUs, to ensure that the communication interface to tags and smartphones is identical.

B. Development of DESFire applet

In 2010, Gemalto announced the world's first implementation of a transport application compliant with the DESFire specification in a SIM (Subscriber Identity Module) or UICC (Universal Integrated Circuit Card) card [8]. This suggests that Gemalto has already developed a Java Card applet to emulate a DESFire tag.

We decided to make our own implementation of the DESFire specification in Java Card, because we wanted to explore different options to visualize the content of the emulated DESFire tag to the user (see Section III). In some approaches for the user interface, we required access to the source code of the DESFire emulator, e.g., when a second applet is used to read the content of the electronic wallet using inter-applet object sharing.

Within the EVENT project Jorge Prado made an initial proof-of-concept implementation of the DESFire specification in Java Card¹ and his results were published in [9]. Afterwards Dries Schellekens independently implemented a DESFire applet, with a clearer structure, a more mature code base and a proper documentation. This second implementation was used in the remainder of the EVENT project and it will be described in this work.

The main design criterium during the design of the applet was compatibility with the DESFire specification. We also tried to write secure and reliable code, that is protected against card tear (by using the transaction functionality of the Java Card runtime environment), and we did some basic performance optimizations (e.g., keeping certain session data in RAM memory). Since the DESFire EV1 specification is pretty extensive and rather complex, the decision was made to only implement the subset of the specification that is useful within our project.

We implemented all the card level commands, which are used for basic configuration of the tag and to create or delete applications on the tag, and all the application level commands, which include the listing of all files within a selected application and the creation and deletion of files.

¹This implementation is available as open source software on <http://code.google.com/p/java-card-desfire-emulation>

At the file level we fully support value files, as these are used to store vouchers in the electronic wallet and partially support standard files,² but we do not support record files. As explained earlier, DESFire support up to 28 applications per card and 32 files per application. In theory we could have supported more applications with our emulator, but we respected this limitation.

On Java Card it is recommended [10] to construct all necessary object instances once during applet installation and to avoid constructing objects dynamically at runtime. Since not all Java Card runtime environments support a garbage collector, one can avoid out-of-memory situations by ensuring no additional objects are constructed at runtime. However, we opted not to initialize the maximum of 28 DESFire applications, 32 files per application and 14 keys per application since in practice only a fraction of these will be used. For this reason we instantiate the application, file and key objects dynamically at runtime. If the specific Java Card runtime environment supports it we ask the runtime environment to garbage collect the objects when they should be removed. If garbage collection is not supported we logically delete the object; in this case they remain in memory but they are no longer accessible.

For compatibility reasons we install the applet using the same Java Card AID as registered by NXP. While selecting an AID on a target before doing a transaction is not strictly necessary on a native DESFire tag, it may be necessary on a secure element where multiple applets are installed or on a secure element where we cannot install an applet as being the default selected applet (e.g., on a UICC the SIM applet is already installed as default selected applet). For this reason our terminals are programmed to always select the DESFire AID before doing a transaction.

DESFire supports two different authentication schemes. The legacy authentication scheme is a challenge-response protocol based on 3DES in a non-standard cipher block chaining (CBC) mode and the standard authentication scheme supports both 3DES and AES in a standard CBC mode. Standard CBC decryption uses the block cipher in decryption mode, but the legacy DESFire CBC decryption uses the block cipher in encryption mode. This signifies that the CBC decryption library from the Java Card runtime environment cannot be used. In our work, we use legacy authentication because this allows for a simpler implementation, but we plan to migrate to standard authentication in the future. Note that a lot of the secure elements that we tested, lack AES support. This is not a problem for our project however, since we are using 3DES, which is supported by most secure elements.

The DESFire specification supports three communication settings, namely plain, MACed and fully encrypted. In the MACed setting, a message authentication code (MAC) value

²In our implementation the size of the standard file is limited.

is appended to every response from the tag. This MAC value is calculated using the CBC-MAC algorithm in the case of a legacy authentication session and the CMAC algorithm for standard authentication sessions. We made an implementation of the CMAC scheme, as this is not supported natively by Java Card. In the fully encrypted setting, a CRC16 or CRC32 checksum is added to the plaintext message for data integrity, and subsequently the message is encrypted. The Java Card architecture supports both these algorithms natively. However, because of endianness issues the CRC16 checksum from Java Card is not compatible with the version from DESFire. Furthermore, most Java Card runtime environments that we tested, do not support CRC32. Therefore we had to make our own table based software implementation for both these checksum algorithms.

While not fully covering the DESFire specification yet we managed to implement the DESFire applet for all the functionalities we need, in about 3600 lines of Java Card code.

C. Deployment possibilities on SE

According to Mobey Forum,³ “The SE is a dynamic environment, where applications are downloaded, personalised, managed and removed independent of each other with varying life cycles.” The requirements are portability (if handset is changed, applications need to be available on new phone again), security (certified by payment industry or trusted third party), multi-application (each application provider has access to its own security domain in the SE), and remote management (download of tickets or top-up over-the-air (OTA) should be possible). The SE can both be accessed by the baseband controller (internal) or by the NFC controller (external).

The biggest issue, and main reason for the relative slow breakthrough of NFC, is who is going to control or manage the SE. There are in fact four possibilities, again all with their advantages and disadvantages:

- **Handset manufacturer centric approach:** In this approach the SE is integrated internally in the phone and the SE is managed by the handset manufacturer. In our opinion this is the least feasible option to deploy our applet, since it is seriously lacking the portability requirement.
- **MNO centric approach:** The Mobile Network Operators (MNOs) are already service providers and can therefore easily add a payment service to their list. In this approach, the SE will reside on the UICC (SIM), which fulfills the portability requirement.
- **Service provider centric approach:** External service providers e.g., Visa and Mastercard are very powerful players in the payment industry and always want their brand to be visible. In this approach, the SE is located

on an external memory chip such as a microSD card or as an active sticker.

- **Neutral third party:** Because of the complexity of the NFC eco-system, a fourth option imposes itself, where an independent third party manages the SEs.

Figure 1 visualizes the three possible locations for a SE in an NFC-enabled smartphone. Additionally there are two more options to add NFC functionality to legacy phones. In the remainder of this section we will discuss these five types of SEs and our experience with deploying the developed DESFire applet on them.

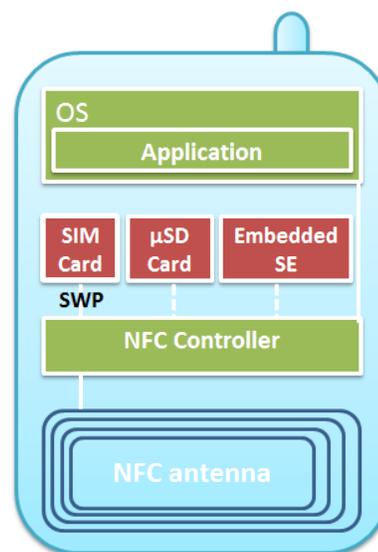


Figure 1. Possible SE locations

1) **Embedded Secure Element:** Smartphones that have a non-removable, integrated secure element, which is connected to the NFC controller, fall into this category. The NXP PN65 chip, which for example is used in the Google Nexus S, the Samsung Galaxy Nexus and the Samsung Galaxy S III devices, even embeds the NFC controller chip (PN544) together with a SmartMX secure element onto one chip. Other smartphones have a separate NFC controller and secure element.

Most NFC phones with an embedded secure element available today cannot be used to deploy the DESFire applet, because the embedded secure element is locked and can only be controlled by the manufacturer. For example on the Google Nexus S and the Galaxy Nexus the secure element is owned by Google and used for its Google Wallet service. The secure element embedded on the Samsung Galaxy SIII on the other hand is owned by Samsung itself. The relatively old Nokia 6212 feature-phone contains an embedded secure element on which we successfully deployed the DESFire applet onto.

2) **Universal Integrated Circuit Card:** Most smartphones already include a secure element in the form of a UICC,

³<http://www.mobeyforum.org/>

which is normally issued by an MNO and contains the SIM applet that enables secure authentication on the mobile network. To be able to use this secure element for NFC applications the SWP (Single Wire Protocol) standard was defined. This standard specifies how the communication between the NFC controller and the UICC works. To be able to use this setup one has to use a SWP compatible UICC card together with an NFC smartphone, which uses SWP to communicate with the secure element.

Most UICC cards issued today by MNOs are not SWP enabled, probably caused by a lack of applications and the increased price compared to regular UICC cards. From Gemalto we received a UICC card, which supports SWP for research purposes. To test this setup we used the Android-based Nexus S smartphone, which supports SWP. When operating in card emulation mode the NFC controller uses the embedded secure element by default.

It is possible however to patch the libnfc-nxp library, which Android uses to communicate with the NFC controller, in such a way that the NFC controller is instructed to use the SWP link to access the secure element, which then is the UICC. After patching the code we can rebuild the Android OS and flash the device with the new ROM.

Initial testing shows that we can access the DESFire applet on the UICC over the NFC interface. There is still a slight problem with the patch that causes the reader to sometimes detect a Felica target instead of the expected ISO14443A target.

We can conclude that while this is certainly not a feasible method for regular user to apply, this shows that it should be possible for a MNO to decide to distribute SWP enabled UICC cards and a selection of patched smartphones to its subscribers.

3) *Secure microSD card*: Secure elements in the form factor of a microSD card are commercially available. Unlike the NFC UICC solution where there is the standardized SWP communication link between the NFC controller and the UICC, there is no such standard communication link to the SD card. There is an effort in Taiwan where a customized HTC device is used that has an SWP-SD link, however this device is not internationally available. In absence of such a direct communication channel between the NFC controller and the secure element a workaround could be developed in the form of a regular application running on the phones operating system, which can access both the NFC controller and the secure element, effectively acting as a proxy relaying APDU messages between both components. Besides the performance being sub-optimal this setup also requires the relaying application to be running constantly, which can on the other hand be more secure for the end user. Since the application can easily be modified extra care must be taken that the system is not subject to man-in-the-middle attacks by ensuring that all data passing through the application is fully enciphered. A secure system should



Figure 2. DeviceFidelity In2Pay

always be insensitive against man-in-the-middle attacks to prevent an attack on the radio interface, however such an attack is harder to achieve in practice than executing the attack against the software component.

4) *NFC-enabled microSD card*: An alternative to this is the use of microSD cards, which have an embedded NFC controller chip that will grant NFC functionality to the host device. The antenna itself can be either external or integrated in the package. Additionally, microSD cards use similar read/write functions as integrated NFC handsets.

Both have their advantages and disadvantages. An NFC microSD card is basically plug and play, thus eliminating difficult setups, but requires the handset to have a microSD slot.

The DeviceFidelity In2Pay product (see Figure 2) provides a microSD based secure element with integrated NFC antenna and external range extender (which is to be glued on the inside of the back cover), which could turn many smartphones with microSD slot into NFC smartphones [11]. The related iCaisse product is a case designed for iPhone. The case contains an NFC antenna, and a slot for the In2Pay microSD card, which together make an iPhone NFC capable.

For Android devices we also have rather bad experiences with the NFC SD card range extender. This should make it possible for an Android smartphone with a microSD slot, to work in the different NFC modes by amplifying the low power RF field generated by the SD card, using two tuned antennas. Even with the supported smartphone types, and matching back covers, the amplifier did not seem to be reliable enough for our use case.

5) *Active NFC bluetooth sticker* [12]: Active NFC stickers provide a way to gain NFC functionality for any Bluetooth enabled smartphone. NFC stickers are contactless cards/tags designed to be glued on the back of a mobile phone and are designed to offer a solution to the

current complex NFC ecosystem that prevents a worldwide integration of NFC technology in mobile handsets. A ferrite backing layer prevents distortion to occur between the components of the phone and its radio signal. The sticker also has an internal antenna installed for communication purposes. The NFC sticker on the other hand only needs a Bluetooth radio, which is a very common feature in most of the mobile phones currently produced. The drawback is however that these are less user friendly to setup compared to NFC MicroSD cards. The sticker has been designed to be as small as possible. A small low voltage battery is used to power the internal Bluetooth chip. This chip is responsible for setting up a connection with the handset and has the capability of making between 300 and 500 connections before the battery is drained. The battery itself can be wirelessly recharged using a specialized USB-charger.

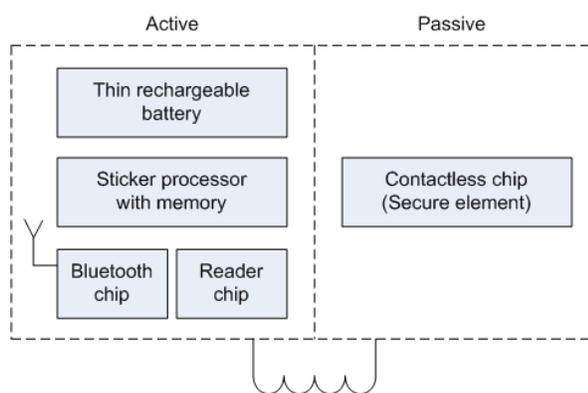


Figure 3. MyMax NFC sticker architecture

The MyMax sticker can act either passively or actively (see Figure 3). An active sticker will rely on its own internal power source, while a passive sticker will use an emitted magnetic field from an external reader to draw power from. Furthermore, the sticker can go into 3 different operation modes; the first mode is a passive mode where the sticker will act as a passive NFC tag. An external reader can be used to read and/or alter the contents of the NFC chip/internal SE on the sticker. It is important to note that a sticker operating in this mode will work completely independent from the handset that it is attached to (i.e., the handset is not required to be powered). The second mode requires that both the MyMax sticker and its corresponding mobile phone draw power actively from an internal power source. This mode allows a connection to be established between the sticker and the mobile phone. Consequently, the content of the internal SE/NFC chip can be read or changed by the handset through this link. The third mode takes advantage of the internal reader chip of the sticker, which makes it possible to create a connection to an external tag and allow the mobile phone to read or change the contents of this tag.

III. USER INTERFACE

Mobile handsets offer a very large advantage in terms of user interactivity when compared to passive tags. The former comes equipped with a functional keyboard and screen and grants the possibility of an interactive interface for the mobile electronic wallet.

A user interface (UI) provides the user a quick overview of the items (both online and offline) available on his electronic wallet. The user should be able to browse through the coupons, loyalty cards etc, and consult the remaining value of the vouchers. Additionally, it should be possible to buy new items or top up existing vouchers OTA using the 3G connection. Ideally this UI should be available across the different major smartphone platforms.

Section III-A provides an overview of the different possibilities for developing a UI that interacts with the wallet on the SE. In Section III-B and III-C we will describe the implementation of two possibilities.

A. Options for the user interface

We identified the following possible interface types: a native application, a SIM Application Toolkit (SAT) applet and a Smart Card Web Server (SCWS) servlet.

A native application for a specific platform like Android has the disadvantage of not being cross-platform, but is the most flexible in terms of interacting with the hardware like the NFC controller. The cross-platform issue can be partly diminished by making use of framework like PhoneGap⁴, which allows the generic UI logic to be written in HTML and JavaScript to allow reuse over different platforms. The platform specific parts like interfacing with the hardware are provided by plugins per platform.

The SAT on the other hand offers more interoperability since everything is stored on a SIM card, which is useable by most handsets and is furthermore deemed to be very secure. While this may sound tempting to use, the downside is that it lacks seriously in terms of visual interface options. Recently more and more smartphones are not including support for SAT interfaces anymore, probably because of the poor usability and the fact that SAT application never were really successful. For these reasons we did not implement a SAT-based solution.

A rather new and promising option is the use of a SCWS installed on a Java Card. A SCWS is a HTTP 1.1 web server embedded on a Java Card and is available on some secure elements since the Java Card 2.2 version, offering a device independent way of the management of personal user data in a secure fashion. This option still puts the application on the SE for security and interoperability purposes. The difference is however that a relatively good HTML interface can be offered by providing static and dynamic content to the browser of a mobile phone through the <http://127.0.0.1:3516/>

⁴<http://phonegap.com/>

address, which is OS independent. If SCWS applications become popular it is to be expected that a closer integration in the smartphone's OS will be provided. For Android a patch [13], which enables the browser to contact the SCWS using the Bearer Independent Protocol instead of the normal TCP/IP stack, is currently required. The application and data can be additionally managed externally through an over-the-air link using web protocols. A secure tunnel will be opened between the SCWS on the Java Card and the OTA platform, which is used for the administration of the SCWS.

The last interface type seems to be the most beneficial in terms of the electronic wallet project, because of its advantages in both maintaining a good visual interface as well as the interoperability, security and user friendliness aspects. However, the disadvantage of the SCWS is that most UICC deployed today do not include support for SCWS. The price of an UICC with support for SCWS is still substantially bigger than a normal UICC. Combined with the fact that there are no compelling SCWS applications yet there is no reason for MNOs to issue the more expensive UICC to its subscriber at the moment. In the next section we will go into more details about a SCWS implementation.

B. User interface based on Smart Card Web Server

The content of the event wallet is displayed in the interface on the mobile phone of the user. This HTML based interface can be divided into a number of interface components that are individually requested from a SCWS residing on the Java Card through various HTTP requests.

The servlet that is responsible for providing the interface with the necessary data is running on the SCWS and will communicate using the Shareable Interface Objects (SIO) mechanism [14], [15] to the DESFire emulator on the Java Card. This emulator is another applet residing on the Java Card and will (after authentication) provide the servlet with the desired data. The servlet will then, based upon the collected data, give an answer to the previously made request by the interface.⁵

Updates to the wallet content on the emulated DESFire card are done mainly through external point of sale terminal, which are available on either the event itself or through an OTA purchase. The design of a mobile interface requires some special attention, more so than its workstation counterpart.

1) *Application interface:* First, there is a large differentiation between different cell platforms, each whom presents its own interface. Secondly, physical obstacles such as different screen sizes, aspect ratios and physical buttons need to be taken into account as well. Lastly, there is the user aspect, which demands that an interface needs to be intuitive and easy to learn.

⁵For a SAT-based UI, the method to access the DESFire applet is the same, since both are Java Card applets running on the same SE as the DESFire applet itself.

We made the decision to use a SCWS to provide a consistent interface by taking full advantage of the mobile phone browser capabilities, which is tasked to render an HTML based interface for the wallet application. This interface is theoretically universally applicable to any mobile phone that supports Java Card. Recent developments like jQuery Mobile⁶ and PhoneGap make it possible to build native looking and responsive HTML and JavaScript-based applications.

The design of the interface of the application itself is based upon known design principles [16], which dictate how information on a page is to be presented to its user. This includes but is not limited to bringing information to the top of the interface by limiting the amount of links a user has to go through thus, minimizing navigation or bringing a collection of relevant information together, based on the desired intent of the user.

2) *Interface structure:* The wallet interface consists of three main tabs that allow a user to browse through a number of subtabs. The main interface tabs are the "Events", "Wallet" and "Settings" tab.

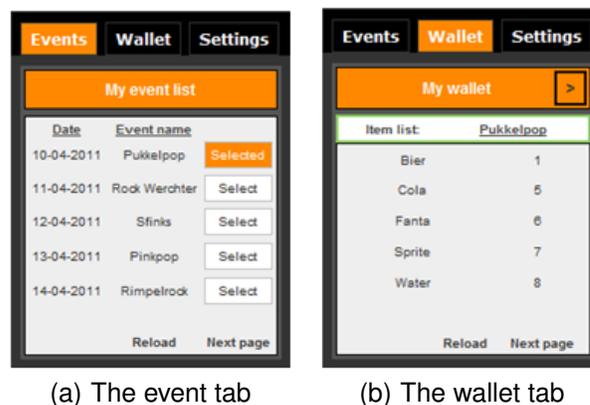


Figure 4. The user interface

The "My event list" subtab (Figure 4(a)) is used to list all the possible events that a user currently has access to. This list is kept up to date through a connection with a backend server. A number of possible items are linked to each specific event and are shown in the "My wallet" subtab (Figure 4(b)). Items can be bought and/or spend using the local terminals or through OTA functionality.

The settings tab allows for various options, including setting the number of listed items and/or events on the events and wallet tab pages. The interface is resolution independent and can thus be used on a number of different handsets. The layout can additionally be changed altogether depending on the preferred style of the user.

Finally, the wallet tab includes the option of purchasing items OTA. The purchased items will be listed on the "My

⁶http://jquerymobile.com/

wallet” subtab as “reserved items”, meaning that they still need to be synchronized by specialized terminals called “sync points”. The main advantage of this is that the wait time for the user at a terminal will be cut down significantly since a user only has to touch the terminal to transfer the previously purchased items over the NFC link.

3) *Data flow*: The next part of the application consists of two elements, namely the GUI Servlet and the DESFire emulator, which are both tasked with the provision of the actual data to the interface mentioned earlier.

A GUI Servlet is a Java Card application that runs on the SCWS and will act both as a server and a client when data is requested by the interface, since it will serve information to the interface after it has requested the necessary data from the DESFire applet.

4) *Retrieval of data*: A backend server forms the backbone of the system, because all the event dates, event names and item names are requested from this server. Providing a page in the interface with values requires the servlet to request a list of AIDs and FIDs first from the DESFire emulator.

These IDs represent the various events and items tied to an event respectively and are required to be translated by the back-end server to their actual corresponding names. The item values shown on the wallet page are collected by using an AID to select a specific event on the DESFire emulator and then request the value of a specific item of that event using its FID.

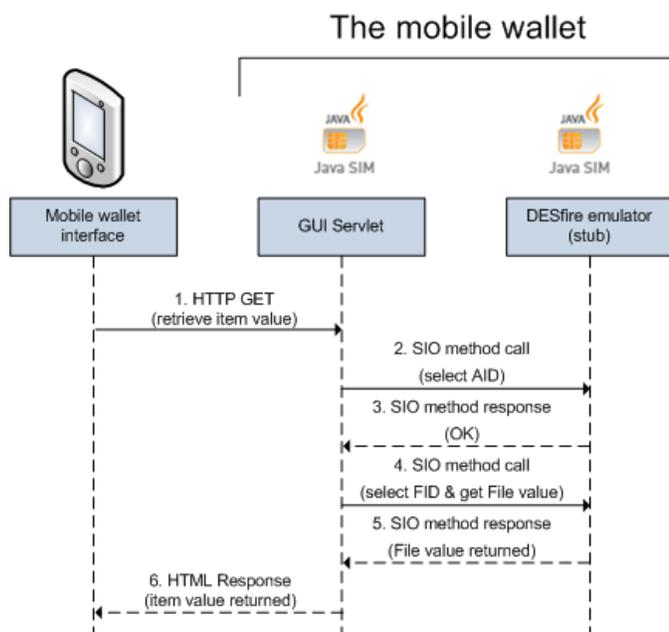


Figure 5. Retrieval of an item value

Every request done by the interface will thus trigger a series of steps in the background of the application

(Figure 5). The GUI servlet will perform a SIO method call to select the AID that is linked to the desired event. The emulator will respond with a confirmation message and the actual file value will then be requested next by using the FID of a specific item. The File value is passed back to the servlet, which will pass it to the interface for visualization.

The backend server and its data have been temporary replaced in our project by a number of vectors that store the different names and dates mentioned before in a hardcoded manner. The reason for this is identical to the stub in that it is caused by the fact that the EVENT project is still in a relatively early stage. We implemented and tested the SCWS using an emulator, since a SCWS enabled SE was not in our possession at the time of writing.

C. User interface based on a native application

The mobile application needs to communicate with the DESFire applet on the secure element using the same APDUs that are used by the terminals. The problem here is that there is not yet a standard API available in the Android SDK to communicate with a secure element from an application. In 2011 Giesecke & Devrient (G&D) proposed patches to the Android Open Source Project that enable a simple interface for a developer list all the supported and available secure elements, and to send APDUs to a secure element [17]. This SmartCard API is an implementation of the SIMalliance Open Mobile API [18] that enables an Android application to communicate with a secure element. This SmartCard API can be extended with plugins (named terminals), which implement support for accessing a specific secure element type. By default terminal types are included for ASSD (Advanced Security SD Card) secure elements, UICC and embedded SmartMX based secure elements. The plugin architecture allows us more secure element types like the MyMax sticker or the DeviceFidelity microSD card to be accessible from applications using the same standard SmartCard API, thus providing an abstraction layer around the specific secure element.

Figure 6 shows the different possibilities of communicating with an SE from an application.

1) *G&D Mobile Security Card*: This SD card based secure element was used to establish a working link with a secure element through the SmartCard API. Once this link was established communication with the DESFire emulator could be done directly, because it was installed as default selected. This setup was tested with a Samsung Galaxy Y, a smartphone without NFC, to get started on the graphical user interface for the application and secure element communication. The terminal interface could not be tested with this secure element, because of the absence of an NFC controller and antenna.

2) *DeviceFidelity In2Pay and iCaisse*: Accessing the In2Pay secure element happens with the In2Pay API that is available for all major smartphone operating systems. We

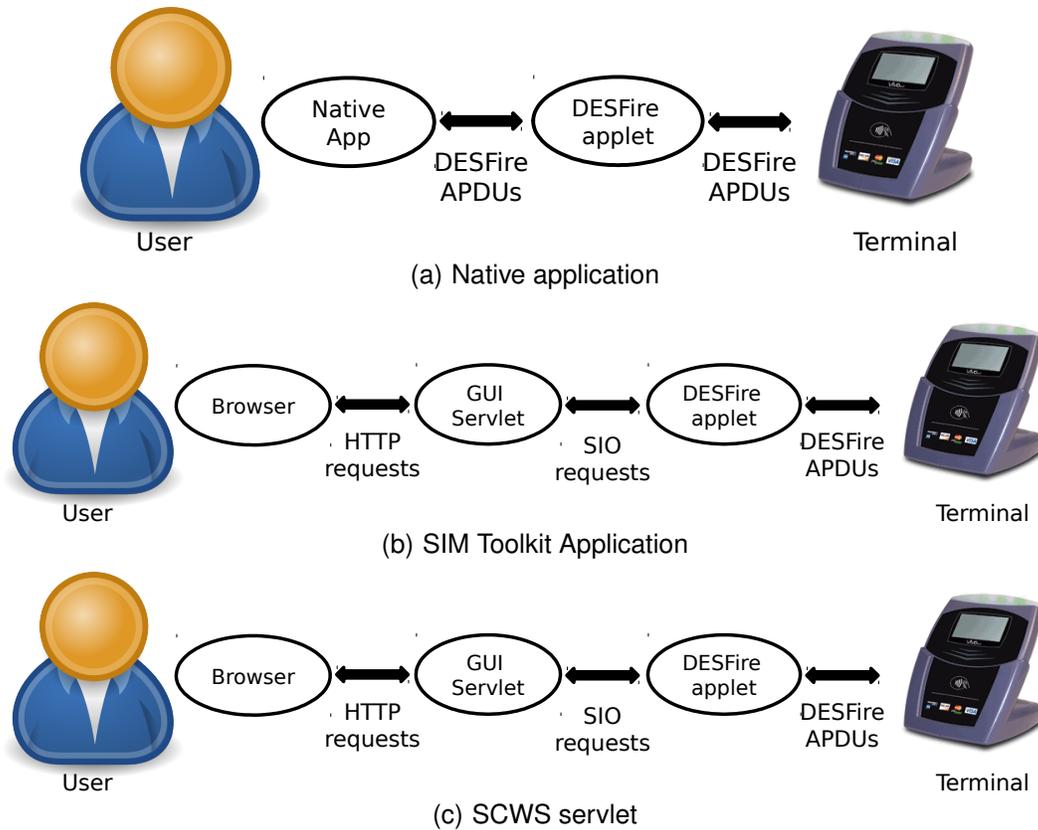


Figure 6. Interface types

tested this on the Android OS, and on iOS (making use of the iCaisse). Communication with the DESFire applet through this API gave us timeout errors on some APDU commands, while accessing the applet over the NFC interface worked as expected.

Looking to resolve this issue we established that communication time with the DESFire emulator on the Device-Fidelity card took longer than when deployed on MyMax NFC sticker, using Jload. The command to retrieve a list of all applications that the DESFire emulator contains, took about 312 ms while deployed on the sticker it took only 140 ms. Similar differences were found for other commands. Remarkable was that a select command on an application on the DESFire emulator took 203 ms and gave no time out error, whereas the command to get all value files from a application, when only one was available, took only 16 ms more. When looking closer we established that the command time to get the file value, was only 94 ms while it took 125 ms to select the corresponding application.

3) *Active NFC sticker*: The Twinlinx MyMax sticker comes with an Android API, which provides applications any easy interface towards the sticker. The API allows the application developer to set up the Bluetooth connection, to connect to the internal SE and transmit APDUs. There

are some usability aspects that cause this solution to be more complex and cumbersome for a user than an integrated solution however. Before the first usage one has to set up the Bluetooth pairing manually. Before executing a transaction the user has to remember to turn the sticker on. The sticker will shutdown automatically after a short period, to conserve battery. We perceived accessing the sticker from the application as rather slow and not user friendly. The next version of the sticker and API will have a feature that allows the phone to vibrate using a certain pattern, which will be detected by the sticker so it can turn itself on. We have not yet received this new version, but this could enhance usability.

4) *UICC*: To build a UI to interact with the wallet stored in the DESFire applet on our UICC, we need to be able to communicate with the UICC using APDUs. Currently there are several obstacles preventing us to do this on Android. The Android operating system runs on the application processor, where the UICC is usually connected to the modem or baseband processor. The OS can communicate with the UICC by using a small set of AT commands that are supported by the baseband processor. The 3GPP 27.007 specification [19] defines the following additional AT commands that enable generic SIM access using APDUs:

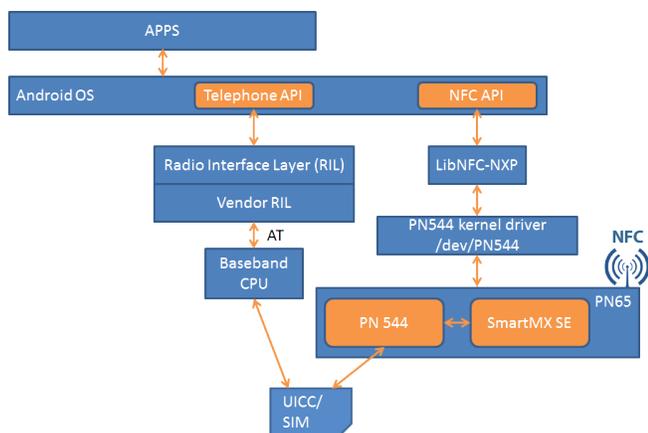


Figure 7. Android UICC access overview

- AT+CSIM: Generic SIM Access
- AT+CCHO: Open Logical Channel
- AT+CCHC: Close Logical Channel
- AT+CGLA: Generic UICC Logical Channel Access

We do not know of an Android device that has a baseband processor supporting the mentioned AT commands at the moment. These changes also need to be made in the Android Radio Interface layer (RIL), this is an abstraction layer between the Android telephony manager services and the baseband modem. RIL exists out of two parts (see Figure 7): the RIL daemon, which is a wrapper to have a standardized interface to the second part, and a vendor specific library. At the moment, only a subset of the SIM ToolKit (STK) [10] is available and transparent APDU communication with the UICC is not allowed. The AT Commands that are needed [20] are specified by the RIL extension specification from G&D [21].

The vendor specific RIL is a closed sourced component that cannot be patched to allow generic APDU access. However, the SmartCard API project contains a patch for the Android emulator that allows applications running in the emulator to access a UICC that is attached over a Personal Computer/Smart Card (PC/SC) interface to the host computer. The SmartCard API also proposes a patch to Androids Telephony API, extending this API with methods for SIM access. Since accessing the UICC using the official RIL way seems not feasible today without extensive manufacturer support we wanted to try to communicate with the UICC via the NFC controller. This would be theoretically possible, by adapting the libnfc-nxp driver, if the NFC controller chip (e.g., PN65) would allow this. We did not succeed in getting access to the PN65 datasheet from NXP however.

IV. CONCLUSIONS

In this paper, we demonstrated that backwards compatibility with a tag-based solution can be accomplished on NFC-enabled smartphones. We have developed a JavaCard

applet that implements the DESFire specification, and we have tested the applet on many different physical SEs. For our use case, which starts from an existing payment infrastructure, no changes were needed to the terminal software, proving the backwards compatibility. Consequently, if NFC technology becomes widely available in the nearby future, the payment infrastructure that is already in place, can be reused. Although no extensive studies were performed, the smartphone solution gives satisfactory results regarding transaction performance.

In contrast, accessing the wallet content with a UI remains a challenge. We have identified three approaches to interact with the DESFire applet. We had limited success with developing a native Android application to access the DESFire applet on an external SE (such as a microSD card or the bluetooth sticker). However, we were unable to test a solution using an embedded SE, because the DESFire applet cannot be deployed on the SE without support of the handset manufacturer. Furthermore, in the case of a UICC-based SE the vendor specific radio interface layer prevents access to a custom applet on the UICC. Again for this solution to work support from a handset manufacturer is required.

The SAT UI approach appears technically feasible on selected devices. However, this text-based solution does not provide a modern UI experience and therefore we did not investigate this option in our work.

A SCWS-based UI seems a promising approach and we showed the feasibility in emulation. However, the commonly available SEs do not yet support SCWS. Additionally, support for SCWS is lacking in today's smartphone OSes.

From a users perspective, the NFC UICC appears the most suitable solution because it allows to transfer the electronic wallet from one phone to another. The same applies to a microSD based solution, but not all devices are equipped with a microSD slot. Embedded SEs lack this transferability feature, which is beneficial as in our use case vouchers are stored offline on the SE; for an online scheme this is not a strict requirement.

Finally, we would like to point out that, while the development of a contactless electronic wallet in a smartphone is technically feasible, the main hurdle preventing market deployment of such a system is the difficult NFC ecosystem. There are too many SE options and it is unclear which stakeholder is in control of the SE.

ACKNOWLEDGEMENTS

This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007), by the Flemish iMinds projects, and by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT-II. In addition, this work was supported by the Flemish Government, IWT SBO MobCom and IWT Tetra EVENT.

The Technology Transfer (Tetra) program of the IWT, the government agency for Innovation by Science and Technology⁷, is meant to help Flemish companies and research centres in realizing their research and development projects. Together with the support and feedback from Antenor, Buzy.be, easyFairs, Events Catering Bevers, Event Drive, First of Kind Solutions, Mobile For, and TwinLinx both research groups managed to finish the EVENT project successfully by giving the partners a well-founded vision on the electronic wallet of the future through live test events in the field.

REFERENCES

- [1] K. De Kock, T. Van Herck, G. Ergeerts, R. Beyers, F. Schrooyen, M. Ceulemans, and L. Wante, "Building the bridge towards an open electronic wallet on NFC Smartphones," in *MOBILITY 2011 - Barcelona, Spain*, 2011.
- [2] N. Forum, "Whitepaper: Essentials for successful NFC Mobile Ecosystems," p. 24, 2008, [accessed 10-July-2011]. [Online]. Available: http://www.nfc-forum.org/resources/white_papers/NFC_Forum_Mobile_NFC_Ecosystem_White_Paper.pdf
- [3] NFCInsight, "NFC Payments Fact Pack," in *2nd Annual NFC Payments Europe 2012 Conference & Exhibition*, 2012, pp. 2–15.
- [4] J. Neefs, F. Schrooyen, J. Doggen, and K. Renckens, "Paper Ticketing vs. Electronic Ticketing Based on Off-Line System 'Tapango'," in *Near Field Communication (NFC), 2010 Second International Workshop on*, april 2010, pp. 3–8.
- [5] G. de Koning Gans, J.-H. Hoepman, and F. D. Garcia, "A Practical Attack on the MIFARE Classic," in *Smart Card Research and Advanced Applications, 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, London, UK, September 8-11, 2008. Proceedings*, ser. Lecture Notes in Computer Science, G. Grimaud and F.-X. Standaert, Eds., vol. 5189. Springer, 2008, pp. 267–282.
- [6] F. D. Garcia, G. de Koning Gans, R. Muijers, P. van Rossum, R. Verdult, R. Wichers Schreur, and B. Jacobs, "Dismantling MIFARE Classic," in *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, ser. Lecture Notes in Computer Science, S. Jajodia and J. López, Eds., vol. 5283. Springer, 2008, pp. 97–114.
- [7] F. D. Garcia, P. van Rossum, R. Verdult, and R. W. Schreur, "Wirelessly Pickpocketing a Mifare Classic Card," in *30th IEEE Symposium on Security and Privacy (S&P 2009), 7-20 May 2009, Oakland, California, USA*. IEEE Computer Society, 2009, pp. 3–15.
- [8] Gemalto, "Worlds First: Gemalto Integrates DESFire Transport Card into NFC Mobile Phone," Feb. 2010, [accessed 12-December-2012]. [Online]. Available: http://www.gemalto.com/php/pr_view.php?id=704
- [9] J. Prado Casanovas and G. Van Damme, "DESfire Emulation Using Java Card," in *Trustworthy Embedded Devices*. Leuven, Belgium: Conference Publishing Services IEEE, 2011, p. 5.
- [10] Gemalto, "Java Card & STK Applet Development Guidelines," 2009, [accessed 10-July-2011]. [Online]. Available: http://developer.gemalto.com/fileadmin/contrib/downloads/pdf/Java_Card_STK_Applet_Development_Guidelines.pdf
- [11] DeviceFidelity, "Mobile Contactless Technology Backgrounder," Jun. 2011, [accessed 12-December-2012]. [Online]. Available: <http://www.devifi.com/assets/whitepaper.pdf>
- [12] TwinLinx, "MyMax2 NFC sticker presentation – Convenience in your hands," Nov. 2011, [accessed 12-December-2012]. [Online]. Available: <http://www.twinlinx.com/upload/file/mymax2presentation.pdf>
- [13] G&D SmartCard API, "Bearer Independent Protocol," 2011, [accessed 10-July-2012]. [Online]. Available: http://code.google.com/p/seek-for-android/wiki/BIP_Extensions
- [14] M. Montgomery and K. Krishna, "Secure Object Sharing in Java Card," pp. 14–14, 1999.
- [15] D. Perovich, L. Rodriguez, and M. Varela, "A Simple Methodology for Secure Object Sharing," p. 7, 2000.
- [16] K. Holtzblatt, "Customer-centered design for mobile applications," in *Personal and Ubiquitous Computing archive, Volume 9 Issue 4, July 2005*, 2005, pp. 227–237.
- [17] G&D, "Secure Element Evaluation Kit for the Android platform," 2012, [accessed 10-July-2012]. [Online]. Available: <http://code.google.com/p/seek-for-android/>
- [18] SIMAlliance, "Open Mobile API Specification," 2011, [accessed 10-July-2011]. [Online]. Available: <http://www.simalliance.org/en/resources/specifications/>
- [19] 3GPP, "3GPP Specification Detail," 2012, [accessed 10-July-2012]. [Online]. Available: <http://www.3gpp.org/ftp/Specs/html-info/27007.htm>
- [20] Telit Wireless Solutions, "AT Commands Reference Guide," 2012, [accessed 10-July-2012]. [Online]. Available: <http://www.telit.com/module/infopool/download.php?id=542>
- [21] G&D, "RIL extension specification," 2012, [accessed 10-July-2012]. [Online]. Available: <http://code.google.com/p/seek-for-android/wiki/UICCSupport>

⁷<http://www.iwt.be/english/welcome>

On-line Safety Monitor Based on a Safety Assessment Model and Hierarchical Deployment of a Multi-agent System

Amer A Dheedan

Department of Computer Science

Delmon University

Manama, Bahrain

amer@delmon.bh

Abstract – The operational safety of critical systems, such as nuclear power plants, aircraft and chemical processes, is typically maintained by the delivery of three real-time safety tasks: fault detection and diagnosis, alarm annunciation and fault controlling. Although current on-line safety monitors play this role to some extent, the problem of consistent and timely task performance is largely unresolved. An aspect of the problem is attributed to the type of monitoring knowledge that informs the real-time reasoning; should it be derived, for example, from off-line design models or the operational context of the monitored system? Another aspect is attributed to whether the monolithic or distributed monitor is able to scale up and cope with the complicated and distributed nature of modern critical systems. To address the problem, this paper develops a distributed on-line safety monitor from monitoring knowledge derived from a safety assessment model of the monitored system and a multi-agent system. Agents are deployed hierarchically according to the architecture of the monitored system and they are provided with portions of the knowledge to reason locally over the conditions of the monitored components and collaborate globally to reason over the overseen behaviour of the entire system. The paper also tests the monitor via an application to an aircraft fuel system and evaluates the approach and results by contrasting them with those of earlier work.

Keywords-*fault detection and diagnosis; alarm annunciation; fault controlling; prognosis; sensory measurements filtration and validation*

I. INTRODUCTION

This article presents an extension of the work that has already been presented in [1].

Dating back to the early 1980s, research effort has focused on the development of advanced computer-based monitors. Since then, computerised on-line safety monitors started to appear as computer systems that are installed in the control rooms of plants and flight decks of aircraft [2], [3].

Computerised monitors have been approached differently in terms of (a) their capacity to deliver three safety tasks: fault detection and diagnosis, alarm annunciation and fault controlling; (b) their architectural nature, monitors could be developed from multi-agent (distributed) or monolithic (centralised) reasoning.

A. Fault Detection and Diagnosis

Fault detection and diagnosis techniques are typically developed as model-based and data-based techniques [4],

[5]. The distinction between these techniques lies in the way of deriving the knowledge that informs the real-time reasoning. Specifically, knowledge of model-based techniques is derived from off-line design models, such as Data Flow Diagrams (DFD), Functional Flow Block Diagrams (FFBD), or more recently from models defined in the Unified Modelling Language (UML). Knowledge about the normal behaviour of the monitored system can be obtained directly from these models. To obtain knowledge about abnormal behaviour, analysis techniques such as HAZard and OPerability study (HAZOP), Functional Failure Analysis (FFA), and Failure Mode and Effect Analysis (FMEA) are used to analyse the design models [6].

Knowledge of data-based techniques, on the other hand, is derived from the on-line context of the monitored system. Knowledge about the normal behaviour is obtained by empirical experiment of fault-free operation of the monitored system. To derive knowledge about abnormal behaviour, possible faults of the basic components are identified (by applying the FMEA to the basic components) and injected experimentally in the operational context. The resulting symptoms and ultimate effects on the functionality of the system are then modelled [7].

In both model-based and data-based techniques, monitoring knowledge is applied to real-time reasoning in executable format as monitoring models. To deliver fault detection and diagnosis, a monitoring algorithm executes the monitoring model by instantiating, evaluating and verifying modelled conditions with real-time sensory data.

Model-based techniques have exploited a wide range of monitoring models, such as Goal Tree Success Tree (GTST) [8], [9], [10], fault trees [11], [12], [13], signed direct graph [14], [15], diagnostic observers [16], [17] and parity equations [18], [19], [20], [21]. Similar variety can be seen with the data-based techniques. Consider, for example, rule-based expert systems [22], [23], [24], [25], qualitative trends analysis [26], [27], [28], artificial neural networks [29], principal component analysis [30], [31] and partial least squares [32].

B. Alarm Annunciation

Alarm is the key means to bring the occurrence of faults to the attention of the operators [33]. Developing an alarm technique involves the consideration of alarm definition, alarm processing, and alarm prioritisation and availability [34], [35].

Alarm definition concerns the definition of mode dependency, which is required to establish a distinction between events that occur due to normal operation and others that occur due to faults, so confusing alarms can be eliminated. State-machines [12], operational sequence diagrams [36] and system control charts [33] are among the models that have been exploited to address this issue. Alarm definition also concerns the definition of an effective threshold, the violation of which would result in verifying the occurrence of an event. Thresholds should not be too sensitive and result in false verification, and at the same time, not too relaxed, which would result in late verification and depriving the operators of knowledge about the actual conditions [35].

In alarm processing, distinction among genuine, consequent and false alarms should be achieved. While genuine alarms should be released, consequent and false alarms should be filtered out to avoid confusing alarm avalanches. Cause-consequent analysis of the design models can establish the distinction between causal alarms that concern the maintenance operators and consequent alarms that concern the pilot operators [37], [38]. Sensory measurement validation can eliminate the potential for false alarms. Recent techniques achieve validation through analytical redundancy among sensors, e.g., see [39], [40], [41], [42]. On the other hand, earlier techniques depended on hardware redundancy [43], i.e., redundant sensors. Although redundancy techniques offer adequate robustness, their applicability is limited since they demand increase in cost, weight and volume.

Alarm prioritisation and availability is the process in which alarms are given priorities according to their importance, so they are selected and announced accordingly [35]. The highest priority is always given to safety consequences [44]. Dynamic and group-presentation are two strategies to prioritise alarms. In dynamic prioritisation alarms might be prioritised by (a) different colours (red, amber, magenta) [35]; (b) different severities, such as catastrophic, critical, marginal and insignificant [45]; (c) presenting the highest priority alarms and hiding and facilitating optional access to the less important ones [35]. Group-presentation takes advantage of the screen display (LCD) to present alarm information in windows according to the hierarchical architecture of the monitored process and the importance of the relevant functionality [46]. Windows may allow operator interaction through facilitating silencing of alarms' sound or suppressing illuminated alarms' lights [47].

C. Fault Controlling

Practically, fault controlling is considered in parallel with the controlling process. Fault controlling is implemented in two different approaches. The first is by manual interference of the system's operators, in which further to the need of an advanced alarm technique, the operators should also be trained and provided with guidance on controlling faults [48], [49], [50].

The other approach is achieved automatically by a computerised controller, which is commonly called a Fault-Tolerant Control System (FTCS) [51], [52]. FTCSs, in turn,

are classified into Active Fault-Tolerant Controlling (AFTC) and Passive Fault-Tolerant Controlling (PFTC) [48].

Research on the AFTC has been motivated by the aircraft flight control system [52]. Faults are controlled by selecting and applying the corresponding corrective procedure. An engine fault of a two-engine aircraft, for example, requires a procedure of: (a) cutting-off fuel flow to the faulty engine; (b) the achievement of cross feed from the tanks that were feeding the faulty engine; (c) applying the corresponding command movements to control the surface and compensational instructions to the operative engine [53].

PFTC relies mainly on redundant components, such as multiple control computers and backup sensors and actuators [54], [55]. Typically, provision of redundant components is implemented by hot or cold standby redundancy. In hot standby redundancy, the system is provided with parallel redundant components, which operate simultaneously (powered up) and each component monitors the output of the other(s). Should any of them fail, the others take over. In cold standby redundancy, only one component is on-line (powered up) and other copies are on standby (powered down). Should the on-line component fail, it is powered down and one of the standby components is powered up by a controller [56].

D. Monolithic and Multi-agent On-line Safety Monitors

Monolithic and multi-agent are two common classes of computerised monitors. The monolithic monitor in [12] has been developed from a monitoring model derived from the application of the Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS) safety assessment technique [57]. The model consists of a hierarchy of state-machines (as a behavioural model) that records the behaviour of the monitored system and its sub-systems and a number of fault trees as diagnostic models that relate detected faults to their underlying causes. The concept was motivated by observation of the fact that immense off-line knowledge ceases its benefit and is rendered useless after certifying the safe deployment of critical systems. The exploitation of that knowledge in the context of on-line monitoring results accordingly in an effective and cost-effective monitoring model.

A quite similar monolithic monitor is developed in [13]. The only difference is that the hierarchy of the state-machine is replaced with the control chart of the monitored system and fault trees are maintained as the diagnostic models.

The main limitation of these monitors is that they are based on a monolithic concept in which all monitoring of a plant is delegated to a single object or device. This does not align well with the distributed nature of most modern systems. Systems are typically implemented as a set of sub-systems, which exist in a complex cooperative structure and coordinate to accomplish system functions. Systems are also typically large and complex and show dynamic behaviour that includes complex mode and state transitions.

As a result, such systems need a distributed mechanism for safety monitoring; first it is essential to minimise the time of on-line failure detection, diagnosis and hazard control;

second, a distributed monitoring scheme can help focus and rationalise the monitoring process and cope with complexity.

In [58] a number of agents are deployed on two levels, lower level and higher level. Each agent is provided with a corresponding portion of the monitoring model; agents of the lower level are provided with functional models, and the higher-level agent has a Markov model. Agents are able to exchange messages to integrate their models and observations and deliver safety monitoring tasks. In a similar concept [59], [60] agents are provided with monitoring models (functional models) and deployed to monitor the deliverable functionality of systems. Agents are also able to collaborate with each other to integrate their models and observations and deliver consistent monitoring tasks.

Multi-agent systems have also been exploited in a different monitoring concept. In [61], for example, a number of agents are deployed to monitor the whole functionality of the monitored system and each agent is provided with a different reasoning algorithm and monitoring model, such as self-organisation maps, principal component analysis, neural network or non-parametric approaches. Agents are also able to collaborate with each other to decide consistently on whether the monitored conditions are normal or abnormal. In [62], a number of agents are also deployed to monitor the entire functionality of the monitored system, but every agent monitors the functionality of the system from different sensory data sources and the same monitoring model and reasoning algorithm, which couples Bayesian network and the method of majority voting.

Despite the monitoring success of multi-agent systems, two limitations have also been highlighted: (a) the typical lack of collaboration protocols that can support effective integration among the deployed agents [63]; (b) the logical omniscience problem in which some monitored conditions may fall beyond the knowledge of the agents [64], [65].

E. Motivation

Despite the above discussed efforts and wide variety of monitoring concepts, still there have been numerous instances of accidents that could have been averted with better monitors. The explosion and fire at the Texaco Milford Haven refinery in 1994, for instance, was attributed to late fault detection, poor alarm presentation and inadequate operator training for dealing with a stressful and sustained plant upset [66]. The Kegworth Air disaster occurred in 1989 because of (a) delay in alerting the pilot of the occurrence of the fault and its underlying causes; (b) ineffective alarm annunciation; (c) the lack of automated fault controlling [67]. Recently, monitoring problems contributed to a fatal accident to Air France flight AF447, in which an Airbus A330 crashed in the Atlantic on 1st of June 2009 and all 228 people on board were killed. The technical investigation partly attributed the accident to late fault detection, misleading alarm annunciation and the absence of clear guidance on emergency conditions, which fell beyond the skills and training of the pilot and co-pilot [68].

Motivated by addressing the monitoring problems of such accidents, this paper develops a distributed safety monitor by synthesising the benefits of two strands. The first

is the exploitation of knowledge obtained from the application of a model-based safety assessment technique (i.e., HiP-HOPS). The second is the distributed reasoning of multi-agent systems. Specifically, the paper looks at:

- The development of an effective formalisation and distribution approach to bring the off-line safety assessment model of HiP-HOPS forward to serve in on-line safety as a distributed monitoring model.
- Addressing limitations that have faced the development of multi-agent monitors. Issues of interest are selecting a suitable reasoning paradigm for the multi-agent system and the development of an effective deployment approach, collaboration protocols and monitoring algorithms.

The ultimate aim is the achievement of a spectrum of monitoring merits ranging from the delivery of effective safety monitoring tasks to the development of a scalable and cost-effective monitor.

The rest of the paper is organised as follows: Section two briefly describes the nature of the monitored system, i.e., modern critical systems. Section three presents the position, role, and constituents of the monitor. Section four tests the monitor through the application to an aircraft fuel system. Section five contrasts the developed monitor and obtained results against earlier work. Section six, finally, draws a conclusion and proposes further work.

II. THE MONITORED SYSTEM

Large scale and dynamic behaviour are two common aspects of modern critical systems, i.e., phased-mission systems. While the former aspect calls into question the ability of the monitor to deliver consistent monitoring tasks over a huge number of components, the latter calls into question the ability of the monitor to distinguish between normal and abnormal conditions. A typical example of such systems is an aircraft, which delivers a trip mission upon the achievement of a number of phases; pre-flight, taxiing, take-off, climbing, cruising, approaching, and landing. Thorough knowledge about the architectural components and the dynamic behaviour is essential to achieve effective monitoring.

To model the mutual relations among the components, a hierarchical organisation is commonly used to arrange them in a number of levels. Across the levels, components appear as parents, children and siblings. Fig. 1 shows a classification of those levels. Levels are classified into three types: the lowest level (level0) is classified as the basic components (BC) level. The intermediate levels extending from level1 to level $n-1$ are classified as sub-system (Ss) levels. The top level (level n) is classified as the system (S) level.

To model the behaviour of the monitored system, it might be required to understand the way in which behavioural transitions are initiated. Typically, transitions are outcomes of, firstly, normal conditions in which the system engages its components in different structures, so it delivers different functionalities. Signals upon which that structure is altered are always initiated by the basic components. For

example, during the cruising of an aircraft, navigation sensors may convey signals to the navigator sub-system (NS), which in turn calculates those signals and notifies the flight control computer (FCC). Assuming that it is time for launching the approaching phase, FCC accordingly instructs the power plant system (PPS) to achieve the required thrust and the surface hydraulic controller (SHC) to achieve the required body motions. The case in which the system uses a certain structure to deliver certain functionality is called a *mode*.

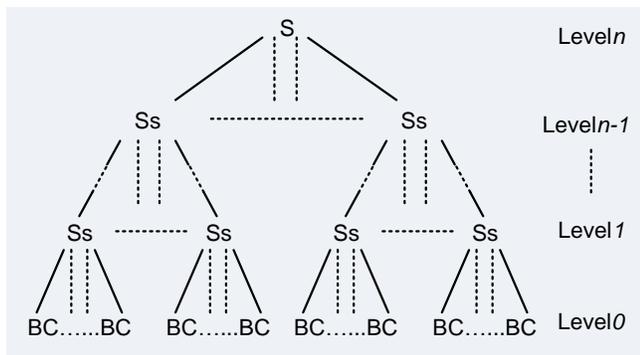


Figure 1. Hierarchical view of the monitored system.

Secondly, dynamic behaviour could be an outcome of the fault or fault tolerating of the basic components. Fault tolerance is typically implemented by two strategies: active fault-tolerant controlling (AFTC) and passive fault-tolerant controlling (PFTC). In the former strategy faults cannot be corrected totally but the consequent effects can be controlled as the system adapts to faults of its components, e.g., the fault of one engine of a two-engine aircraft can be compensated by the other engine. In the latter strategy the system has the ability to tolerate the fault for a while, e.g., faults that are caused by software error, ionisation radiation, electromagnetic interference, or hardware failure can be corrected within a short interval by restarting the relevant component or by isolating the faulty component and starting up a redundant one.

It could, therefore, be said that during a mode, a system may appear in different health *states*, which can be classified into two types. The first is the *Error-Free State (EFS)* in which the system or a sub-system functions healthily. The second type is the *Error State (ES)*, which in turn is classified into three different states:

- *Temporary Degraded or Failure State (TDFS)* in which there is one or more functional failure, but corrective measures can be taken to transit to another state;
- *Permanent Degraded State (PDS)* in which an uncontrollable fault occurs, but the safe part of the functionality can be delivered;
- *Failure State (FS)* in which the intended function is totally undelivered.

Events that are initiated by the basic components play a key role in making the behaviour of a system dynamic. To

track the behaviour, such events should be continuously monitored. Thus, the best hierarchical level to monitor these events should be identified. Fig. 2 illustrates the relationships among the architectural levels and three factors based on which that level can be decided; early fault detection and diagnosis, computational cost and behavioural understanding. Achieving trade-off among these factors could help effectively in identifying the targeted level.

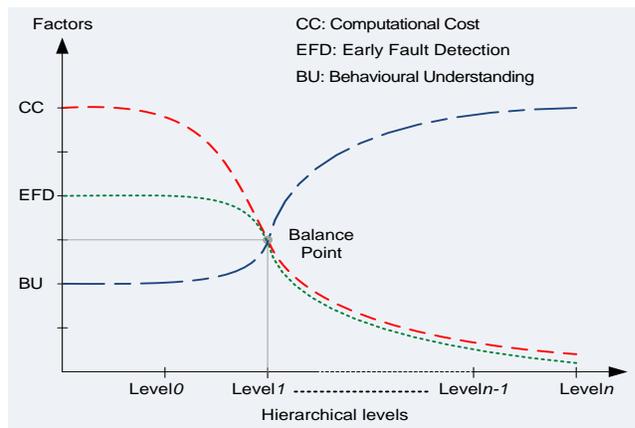


Figure 2. Three monitoring factors and architectural levels.

At level *l* the occurrence of events could be identified as either normal or abnormal, e.g., the decreasing of velocity and altitude seems normal when the flight control computer has already launched the approaching of the aircraft. Excluding knowledge about the modes and focusing only on the measurements provided by the relevant sensors would certainly result in misinterpreting system behaviour, i.e., decreasing velocity and altitude would appear as a malfunction and a misleading alarm would accordingly be released. Having that fact, level *l* would also be the best level – rather than any higher level – since at that level a malfunction is detected while in its early stages. Finally, due to the potentially huge number of the basic components, monitoring events at level 0 is computationally expensive or even unworkable, whereas level *l* offers the required rationality. Without loss of generality, it is assumed that primary detection of the symptoms of failure occurs at level *l*.

III. DISTRIBUTED ON-LINE SAFETY MONITOR

The monitor takes a position between the system and the operators' interface. During normal conditions, the monitor provides simple feedback about those conditions. The monitor plays its role during abnormal conditions, which are triggered by and follow the occurrence of faults. It delivers three safety tasks; prompt fault detection and diagnosis, alarm annunciation and fault controlling.

Prompt fault detection and diagnosis refers to the timeliness of detecting faults while in their early stages and before they develop into real hazards, in parallel with diagnosing the underlying causes. This is supported by

selecting an appropriate hierarchical level at which efficient monitoring of the operational parameters can be achieved in addition to setting and monitoring those parameters against well-defined thresholds.

The task of effective alarm annunciation involves defining thresholds whose violation represents actual deviations of the monitored parameters. It also involves suppressing unimportant and false alarms whose release would overwhelm and confuse the operators. This is achieved by the following:

- Tracking the behaviour of the monitored system and distinguishing among the occurrence of normal, corrective and failure events.
- Releasing an alarm only on the occurrence of genuine failure events and not on other events, such as consequent, precursor or causal events.
- Developing techniques to filter out and validate the sensory measurements. Prioritising alarm presentation is also important to deliver effective alarm annunciation. This can be achieved by distinguishing the important alarms by using different colours, vibration or alerting sounds, and hiding the presentation of the less important alarms, e.g., optional access to the diagnostics list on the operators' interface.
- Annunciation of effective alarm information that could help the operators to direct the system effectively in the presence of faults and control abnormal conditions. Information is presented as (a) assessment of the operational conditions following the occurrence of the fault; (b) guidance on the corrective actions that should be taken manually by the operators; (c) timely prognosis of the future effects of the occurred fault. In order to avoid overwhelming the operators, prognoses would be presented in a timely manner and in the context of behavioural transitions of the monitored system.

The monitor can achieve both active and passive fault-tolerant controlling and also support manual fault controlling by assessment, guidance and prognoses to control abnormal conditions that may fall beyond the trained skills of the operators.

The monitor consists of two main elements. The first is a distributed monitoring model that is derived from an off-line HiP-HOPS safety assessment model, which consists of a behavioural model as a hierarchy of state-machines and fault propagation models as a number of fault trees (Fig. 3). To bring the assessment model forward to serve the on-line monitoring, the achievement of two processes is needed. The first is formalising events that trigger transitions in the behavioural model and symptoms that associate the error propagation paths of faults as monitoring expressions. Hence, the occurrence of events and symptoms can be verified computationally by instantiating and evaluating monitoring expressions based on real-time conditions. Verification of events supports tracking the behaviour of the monitored system and verification of symptoms supports

tracking the error propagation path from the detected faults at level I towards the underlying causes at level 0 .

The second process is distributing the model into a number of models without violating the integrity and consistency of the encoded knowledge; for each sub-system there will be a monitoring model and a model for the entire system appears at level n .

The second element is a multi-agent system, which is a set of Belief-Desire-Intention (BDI) agents. Agents are deployed and provided with their portions of the monitoring models to reason locally at the sub-systems level and also provided with collaboration protocol to integrate globally at the system level and deliver the three safety tasks.

A. Monitoring Expressions

In its simple form, a monitoring expression appears as a constraint that consists of three main parts: (a) an observation, which is either a state of a child or the parent or sensory measurement defined by the identifier of the relevant sensor; (b) a relational operator – equality or inequality; (c) a threshold whose violation results in evaluating that expression with a true truth value, i.e., the relevant event or symptom occurs. Thresholds might appear as a numerical or Boolean value.

The formalisation of events in the state-machine of level I 's sub-systems and the symptoms of the diagnostic model might require more complicated forms of constraint that incorporate (a) observations that should be calculated over a number of sensory measurements; (b) two operational operators, when the threshold is a range of values rather than a single value; (c) a threshold that represents a sensory measurement or a calculation of more than one measurement. Moreover, observations and the threshold might be calculated to find the average of the change of a quantity over an interval (Δt), i.e., differentiation, or the volumes from different sensory measurements at definite timings, i.e., integral calculus. Consider, for example, an expression to monitor a structural leak of a tank of the aircraft fuel system, a case study presented in this paper (Fig. 13). Assuming that the leak is in the inner tank of the left-wing (LW) sub-system, the monitoring expression can be formalised as follows:

$$LL1(T-5) - LL1(T) > \int_{T-5}^T (FL1(t) + FL2(t)) dt + 0.06 \quad (1)$$

where

$(LL1(T-5) - LL1(T))$: is the reduction of fuel level in the inner tank over an interval extending from $T-5$ in the past to current time T .

$\int_{T-5}^T (FL1(t) + FL2(t)) dt$: is the total amount of fuel that has been (a) drawn from the inner tank by pump PL1 over an interval extending from $T-5$ in the past to current time T ; (b) drawn or added by pump PL2 over the same interval. The interval is defined as

5 seconds as the shortest time to detect the structural leak.

0.06: is the maximum allowable discrepancy between the two above observations in normal conditions.

The calculation and evaluation of such an expression necessitate holding sensory measurements over time, i.e., historical measurements. Fig. 3 shows multi-measurement buffer along with its systematic updating.

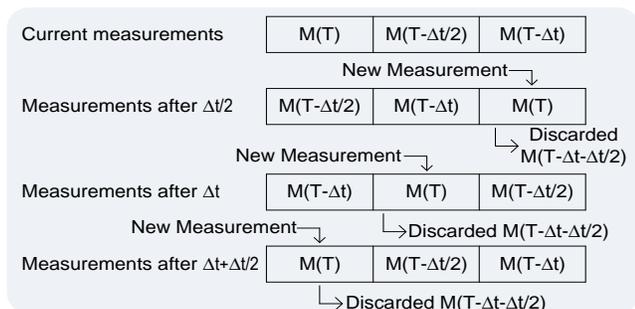


Figure 3. Systematic update of a multi-measurement buffer.

In Fig. 3, the updating process is applied continuously over time and after every elapsing of $\Delta t/2$. It can be seen how the updating maintains a systematic interval of $\Delta t/2$ among the measurements and replaces measurements that fall out of Δt . This structure could hold sensory measurements that suffice for the calculation and evaluation of expressions like expression (1).

Sensors may deliver temporary spurious measurements because of (a) additive white Gaussian noise, such as electromagnetic interference, ionisation radiation and thermal noise; (b) mode changes, which would typically be followed by an interval of unsteady behaviour. The best way to filter out such measurements is perhaps by forming a *timed expression*. Such an expression is evaluated successively over a filtering interval and based on a number of measurements. The final evaluation result is obtained by making accumulative conjunctions among the successive evaluations. If the final result is true, that means the delivered measurements remain the same over the filtering interval. Hence, the occurrence of that event should be verified. The filtering interval of every expression is defined by examining, firstly, the conditions that may result in spurious measurements; secondly, the time intervals at which the involved sensors are demanded – by the monitor – to deliver sensory measurements. For example, in the fuel system case study, monitoring the fuel flow to the port engine requires formalising a timed expression as follows:

$$|FF1| < 0.03 \text{ for } 4 \text{ sec} \quad (2)$$

In practice, sensors may fail permanently and deliver spurious measurements that persist over or even go beyond the filtering interval. In addition to misleading the monitor; such measurements could also affect the controller of the monitored system and result in hazardous failures. Sensory

measurements should, therefore, be validated and faulty sensors should be detected, diagnosed and controlled.

To achieve that, a technique of formalising special monitoring expressions is developed. The technique is based mainly on the sub-grouping approach of [39] and the Sensory Failure Diagnosis Tree (SFDT) approach of [41], [42]. Drawing from the sub-grouping approach, sensors that can detect each other’s faults are identified and based on the idea of SFDT the proper expression is formed.

For example, a sensory failure of the flow meter FC1 of the fuel system can be detected and diagnosed by the following expression:

$$\begin{aligned} & (FC1 > R/7 + 0.03 \text{ for } 6 \text{ sec AND } FC1 > FC2 + 0.03 \text{ for } 6 \text{ sec}) \\ & \text{OR} \\ & (FC1 < R/7 - 0.03 \text{ for } 6 \text{ sec AND } FC1 < FC2 - 0.03 \text{ for } 6 \text{ sec}) \end{aligned} \quad (3)$$

To control sensory failures, the technique suggests isolating the faulty sensor by ignoring its measurements and measuring the same trend from an alternative sensor or from a number of sensors whose measurements can be calculated to correspond as an alternative to the isolated measurement. In the case of isolating the flow meter FC1, the alternative sensor can be the other flow meter FC2.

Extended-Backus Naur Form (E-BNF) notation is exploited to define a general grammar to formalise different monitoring expressions according to the nature of the monitored conditions. In that grammar a set of primitives has been introduced to allow expressions to reference historical values and calculate different monitoring trends. Primitives include a historical operator $S_{ID}(\Delta t)$, which returns historical sensory measurement collected in the past at current time T minus Δt , i.e., T- Δt . Primitives also define more complicated operators, such as the differentiation $D(\text{expression}, \Delta t)$, integration $I(\text{expression}, \Delta t)$, variation $V(\text{expression}, \Delta t)$ and timed expression $T(\text{expression}, \Delta t)$.

By these primitives, monitoring expressions can be presented in standard computation forms. Consider, for example, expression (1); it can be presented as:

$$V(LL1_L, 5) > I(FL1_F + FL2_F, 5) + 0.06$$

Expression (2) can be presented as:

$$T(|FF1_F| < 0.03, 4 \text{ sec})$$

Expression (3) can be presented as:

$$\begin{aligned} & (T(FC1 > R/7 + 0.03, 6) \text{ AND } T(FC1 > FC2 + 0.03, 6)) \\ & \text{OR} \\ & (T(FC1 < R/7 - 0.03, 6) \text{ AND } T(FC1 < FC2 - 0.03, 6)) \end{aligned}$$

A three-value technique: ‘True’, ‘False’, and ‘Unknown’, is also employed to save evaluation time and produce earlier results in filtering measurements and in the context of incomplete sensory data without violating the evaluation logic. Consider, for example, the following expressions:

$$\text{Expression OR } T(\text{Expression}, \Delta t) \quad (4)$$

$$\text{Expression AND } T(\text{Expression}, \Delta t) \quad (5)$$

Evaluating expressions (4) or (5) may require waiting time equal to Δt , i.e., until evaluating $T(\text{Expression}, \Delta t)$. However, the *Expression* part of either (4) or (5) can be evaluated instantly. Hence, knowing that the disjunction of *True* with *Unknown* is *True* and the conjunction of *False* with *Unknown* is *False*, both (4) and (5) can be evaluated instantly. Therefore, in cases in which *Expression* of (4) is evaluated with *True* and *Expression* of (5) is evaluated with *False*, both (4) and (5) could be evaluated instantly with values *True* and *False*, respectively.

B. Distributed Monitoring Model

In the light of the intended three monitoring tasks, agents should be able to track the behaviour of the monitored components over different states, i.e., error-free states (EFSs) and error states (ESs). This is important to distinguish between normal and abnormal conditions and provide the operators with information that confirms whether the conditions are normal or not. In abnormal conditions, agents should provide alarm, assessment, guidance, diagnostics and prognoses. Agents should also have a reference to apply corresponding corrective measures for every fault.

Fig. 4 shows an illustrative view of the HiP-HOPS model. The model is a composite of a behavioural model and fault trees. The behavioural model is a hierarchy of state-machines that captures the behaviour of the system and its sub-systems. Each fault tree records the possible symptoms, propagation paths and underlying causes of a failure event.

Relationships among the components are implemented in the state-machine hierarchy as parent and children components. In the state-machine of the sub-systems of level l , events are originated by (a) the BCs of level $l-1$, which might be failure, corrective or normal events; (b) parent states, such as the error-free state of a new mode of the parent or error states.

In the state-machine of a sub-system of the levels extending from level l to level $n-1$ events appear as error-free and error states of the parent and children. Finally, in the state-machine of the system (level n) events appear as error-free states and error states of the children.

Similarly, error states of the children could also trigger transitions in the state-machines of the parents and vice versa. For example, the failure state of an engine of a two-engine aircraft triggers a transition to the permanent degraded state in the state-machine of the power plant system. The degraded state, in turn, triggers a transition to a new error-free state of the operative engine in which the lost functionality of the faulty engine is compensated.

To distinguish between normal, fault and corrective events, the principle is applied that an alarm should be released on the occurrence of failure events only. Thus,

corresponding alarm clauses should be associated with the failure events of level l , the level at which events are monitored. Computationally, if an occurred event is associated with a “none” then it is either a normal or a corrective event; on the contrary, any other clause means that it is a failure event and the associated clause should be quoted and released as an alarm. While assessment is a description of the given conditions and guidance is about the best actions to be applied in those conditions by the operators, their clauses should thus be enclosed by the states.

To find the appropriate place for incorporating corrective measures, further consideration of the nature of those measures is needed. Typically, there are two different types of corrective measures. The first should be taken after diagnosing the underlying causes. This is appropriate when the verified failure event can be caused by multiple faults of the basic components. Measures to correct any of those causes vary from one cause to another. Measures should, therefore, be incorporated in the diagnostic model (e.g., fault tree), precisely in association with the potential causes.

The second type of corrective measures should be taken at level l , when level l 's sub-systems supported by higher level components (sub-systems or system) apply measures to respond to deviations that have a clear cause. At level l , corrective measures are mostly applied with directions coming from higher levels. For example, in modern aircraft switching to the backup computer sub-system at level l is instructed directly by the flight control system (FCS) at level $l+1$, whenever the primary computer sub-system at level l fails. The instructions are implemented at level l by switching the primary computer off and backup computer on. Measures should also be taken at level l , when level l 's sub-systems supported by level $l-1$'s basic components apply measures to respond to deviations that have a clear cause. Expression (3), for example, relates a failure event of the condensing sub-system directly to a fault of the flow meter (FC1). In this case, measures are taken to isolate the FC1 and depend alternatively on the measurements obtained from another flow meter FC2.

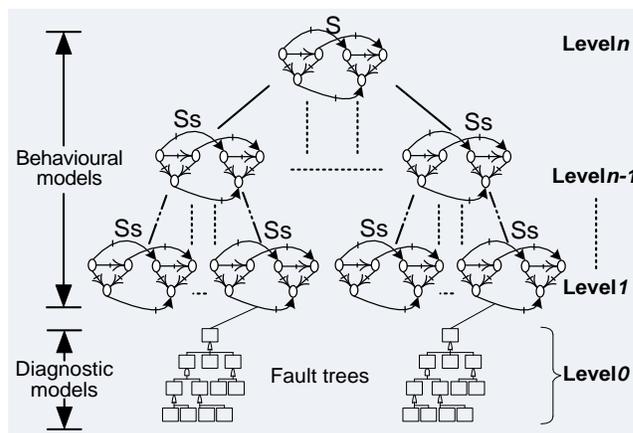


Figure 4. An illustrative view of the HiP-HOPS safety assessment model.

To present the graphical state-machines in an executable format, state-transition tables represent a classic choice. A state-transition table is usually defined as an alternative and formal form to present graphical state-machines and it typically offers the required capacity and flexibility to incorporate knowledge about the operational conditions [69].

Fig. 5 shows an excerpt of the state-machine of level1 of the fuel system case study. Table I shows the state-transition table of the fuel system. It can be seen how the trigger events of the state-machine (Fig. 5) are formalised as monitoring expressions in Table I. For example, event CM_FS of EF, which is the failure state of the engine feed (EF) sub-system during the consumption model (CM) of the fuel system, is formalised as EF_CM_FS == true.

Fig. 6 shows an excerpt of the state-machine of the EF sub-system. It can be seen how the states of the fuel system and its sub-system appear mutually as trigger events in each other's state-machines. Table II shows the formal behavioural model of the engine feed sub-system.

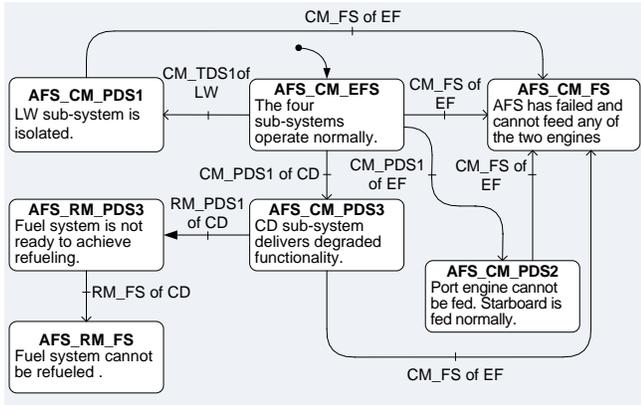


Figure 5. An excerpt of the state-machine of the aircraft fuel system.

TABLE I. STATE-TRANSITION TABLE OF THE AIRCRAFT FUEL SYSTEM.

CURRENT STATE	CONDITIONS	EVENT	NEW STATE
AFS_CM_EFS	Assessment: the four sub-systems operate normally. Guidance: none.	EF_CM_FS == true	AFS_CM_FS
		LW_CM_TDS1 == true	AFS_CM_PDS1
		EF_CM_PDS1 == true	AFS_CM_PDS2
		CD_CM_PDS1 == true	AFS_CM_PDS3
AFS_CM_PDS1	Assessment: LW sub-system is isolated. Guidance: none.	EF_CM_FS == true	AFS_CM_FS
AFS_CM_PDS2	Assessment: port engine cannot be fed, whereas starboard engine is feeding normally. Guidance: none.	EF_CM_FS == true	AFS_CM_FS
AFS_CM_PDS3	Assessment: CD sub-system delivers degraded functionality. Guidance: none.	CD_RM_PDS1 == true	AFS_RM_PDS3
AFS_RM_PDS3	Assessment: fuel system is not ready to achieve refuelling. Guidance: none.	CD_RM_FS == true	AFS_RM_FS
AFS_CM_FS	Assessment: AFS has failed and cannot feed any of the two engines. Guidance: none	none	none

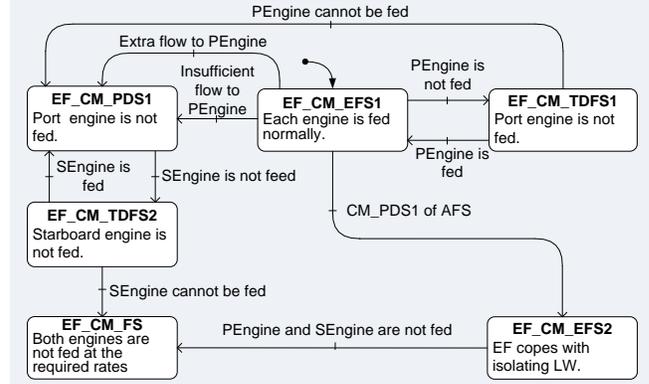


Figure 6. An excerpt of the state-machine of the engine feed sub-system.

Table II differs from Table I, as it incorporates three extra columns: alarm, controlling and diagnosis. The alarm column holds clauses that can be announced to alert the operators. The controlling column may hold corrective measures or “after diagnosis” based on the nature of those measures. The diagnosis column holds the status of whether the occurred failure event has a clear cause or a diagnostic process is needed.

TABLE II. STATE-TRANSITION TABLE OF THE ENGINE FEED SUB-SYSTEM.

CURRENT STATE	CONDITIONS	EVENT	ALARM	CONTROLLING	DIAGNOSIS	NEW STATE
EF_CM_EFS1	Assessment: each engine is fed normally. Guidance: none	T(FF1_F < 0.03, 4);	port engine is not fed	after_diagnosis	needed	EF_CM_TDFS1
		T(FF1_F > R + 0.03, 6);	port engine fed with extra rate	- PF1 = 0; - VF1 = 0;	not_needed	EF_CM_PDS1
		T(FF1_F < R - 0.03, 6);	port engine is fed with insufficient rate.	- PF1 = 0; - VF1 = 0;	not_needed	EF_CM_PDS1
		AFS_CM_PDS1 == true;	none	none	not_needed	EF_CM_EFS2
EF_CM_TDFS1	Assessment: port engine is not fed and recovery is in progress. Guidance: watch for further feedback.	T(FF1_F < 0.03, 4);	feeding port engine cannot be recovered.	- PF1 = 0; - VF1 = 0; - VF2 = 0;	not_needed	EF_CM_PDS1
		T(FF1_F - R < 0.03, 4);	none	none	not_needed	EF_CM_EFS1
EF_CM_PDS1	Assessment: port engine is not fed. Guidance: none.	T(FF2_F < 0.03, 4);	starboard engine is not fed	after_diagnosis	needed	EF_CM_TDFS2
EF_CM_EFS2	Assessment: EF sub-system copes with isolating LW sub-system. Guidance: none.	T(FF1_F < 0.03 AND FF2_F < 0.03, 4);	Both engines are not fed	impossible	needed	EF_CM_FS
EF_CM_TDFS2	Assessment: starboard engine is not fed and recovery is in progress. Guidance: watch for further feedback.	T(FF2_F < 0.03, 4);	feeding starboard engine cannot be recovered.	impossible	not_needed	EF_CM_FS
		T(FF2_F - R < 0.03, 4);	none	none	not_needed	EF_CM_PDS1
EF_CM_FS	Assessment: both engines cannot be fed. Guidance: none.	none	none	none	not_needed	none

During the monitoring time, agents cyclically monitor events whose occurrence triggers transitions from the current state; every cycle is called a monitoring cycle. As such, the computational load of the agents would be less and prompt responses to the occurrence of the events would be established.

A diagnostic process is needed when a failure event and its underlying cause are in a one-to-many relationship. Therefore, a diagnostic model that could relate such events to their underlying cause is needed. As shown in Fig. 4, the HiP-HOPS model incorporates fault trees that can relate functional failures to their underlying causes. More specifically, for every functional failure, which may have multiple causes, there is a fault tree.

On the contrary, when the failure and its cause are in a one-to-one relationship, the name of the cause is stated in the state-transition table of level1's.

Functional failures are related to their fault trees as every failure appears enclosed by the top node of the relevant fault tree. For example, the underlying cause of the failure event "PEngine is not fed" (shown in Fig. 6 and Table II) can be diagnosed by traversing the relevant fault tree, which is shown in Fig. 7. Fig. 8 shows the formal form of the diagnostic model that can be derived from the fault tree of Fig. 7.

Agents initiate the monitoring process by traversing, interpreting and uploading the state-transition tables and diagnostic models into interrelated data structures. Structure type and arrays are declared for this purpose. Arrays support direct addressing of the structures that hold the knowledge, so fast access during the monitoring time is established.

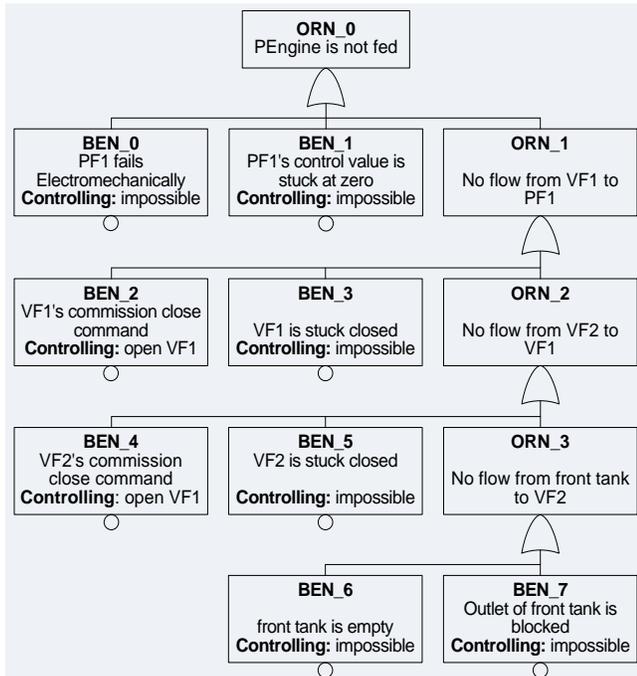


Figure 7. Fault tree of event "PEngine is not fed".

NodeName: ORN_0.
Symptom: T(|FF1| < 0.03, 4 sec).
ChildName: BEN_0.

NodeName: BEN_0.
Symptom: |PF1| <= 20.
Fault: PF1 fails electromechanically.
Controlling: none.
Sibling: BEN_1.

NodeName: BEN_1.
Symptom: |PF1| <= 20.
Fault: PF1's control value is stuck at zero.
Controlling: none.
Sibling: ORN_1.

NodeName: ORN_1.
Symptom: none.
Child: BEN_2.
Sibling: none.

NodeName: BEN_2.
Symptom: VF1 == 0.
Fault: VF1's commission close command.
Controlling: VF1_C = 1.
Sibling: BEN_3.

NodeName: BEN_3.
Symptom: VF1 == 0.
Fault: VF1 is stuck closed.
Controlling: none.
Sibling: ORN_2.

NodeName: ORN_2.
Symptom: none.
Child: BEN_4.
Sibling: none.

NodeName: BEN_4.
Symptom: VF2 == 0.
Fault: VF2's commission close command.
Controlling: VF2 = 1.
Sibling: BEN_5.

NodeName: BEN_5.
Symptom: VF2 == 0.
Fault: VF2 is stuck closed.
Controlling: none.
Sibling: ORN_3.

NodeName: ORN_3.
Symptom: none.
Child: BEN_6.
Sibling: none.

NodeName: BEN_6.
Symptom: (VF2 == 1) AND (VF1 == 1) AND (PF1 > 20).
Fault: front tank outlet is blocked.
Controlling: none.
Sibling: BEN_7.

NodeName: BEN_7.
Symptom: LF1 < 0.1.
Fault: front tank is empty.
Controlling: none.
Sibling: none.

Figure 8. Formal diagnostic model of the fault tree of Fig. 7.

C. Multi-agent System

In addition to the common ability of intelligent agents to achieve integrated reasoning among distributed processes [70], two more reasons underpin the particular adoption of BDI agents as monitoring agents. Firstly, as the reasoning

model of these agents is based on human reasoning, effective automation of the crucial responsibilities of system operators can be facilitated. Secondly, the informative communication as well as the semi-independent reasoning of the BDI agents can support effective collaboration and integration of two different deployment approaches. The first is spatial deployment in which agents are installed on a number of distributed computational machines. Such deployment is needed when the sub-systems of the monitored system are distributed over a geographical area, e.g., a chemical plant. The second approach is semantic deployment in which monitoring agents are installed on one computational machine. Such deployment is appropriate when the sub-systems of the monitored system, although distributed, are close to each other, e.g., an aircraft system.

Fig. 9 shows a general illustration of the monitoring agent. By perceiving the operational conditions and exchanging messages with each other, each agent obtains the up-to-date belief, deliberates among its desires to commit to an intention and achieves a means-ends process to select a plan, which is a course of actions. The selected plan is implemented, as actions towards achieving the monitoring tasks locally and as messages sent to other agents towards achieving global integration. Upon having a new belief, an agent achieves a reasoning cycle; deliberation and means-ends process.

Agents are deployed over the sub-systems and the system, and appear as a number of sub-system monitoring agents (Ss_MAGs) and a system monitoring agent S_MAG, as shown in Fig. 10. Each Ss_MAG of level l updates its belief base by perceiving (a) its portion of the monitoring model, which consists of a state-machine and a set of fault trees; (b) sensory measurements that are taken to instantiate and evaluate monitoring expressions; (c) messages that are received from the parent to inform the Ss_MAG about the new states and messages from siblings, in which they either ask for or tell the given Ss_MAG about global measurements; agents might need to share measurements globally. The main desires of an Ss_MAG of level l are to monitor the local conditions of the assigned sub-system and to collaborate globally with its parent and siblings. On the achievement of the local desire, the intentions are to track the behaviour of the sub-system and to provide the operators with alarms, assessment, guidance, diagnostics, prognoses and control faults. On the achievement of the global desire, the intentions are to exchange messages to inform the parent about the new states and to tell or ask the siblings about global measurements.

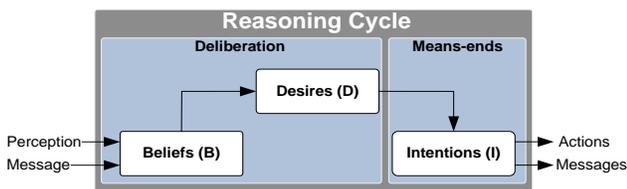


Figure 9. Reasoning cycle of the BDI agent.

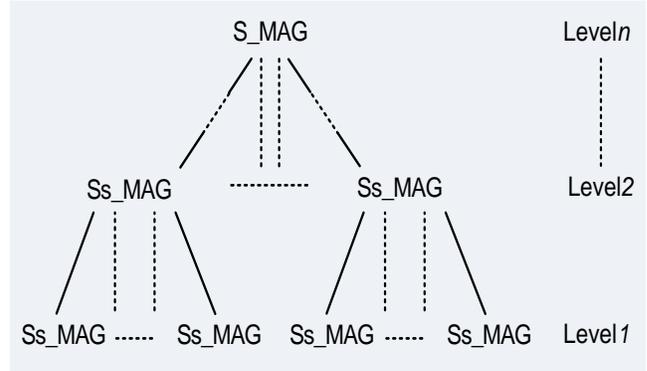


Figure 10. The hierarchical deployment of the monitoring agents.

Each Ss_MAG of the intermediate levels (levels extending from level 2 to level $n-1$) updates its belief by (a) perceiving its own portion of the monitoring, which consists of a state-machine of the assigned sub-system; (b) messages received from the parent and the children to inform it about their new states. The main desires of each of these Ss_MAGs are to monitor the local conditions of the assigned sub-system and to collaborate globally with its parent and child agents. On the local desire, the intentions are to track the behaviour of the sub-system and to provide the operators with assessment, guidance and prognoses of their levels. On the global desire, the intention is to exchange messages with the parent and child agents to inform each other about their new states.

The perceptions, desires and intentions of the S_MAG are similar to those of the Ss_MAGs of the intermediate levels. The only difference is that S_MAG has no parent to exchange messages with.

According to the Prometheus approach and notation for developing multi-agent systems [71], Fig. 11 shows the collaboration protocols among agents to track the behaviour of the monitored system. Fig. 12 shows the collaboration protocol among the Ss_MAGs of level l in which they share their sensory measurements.

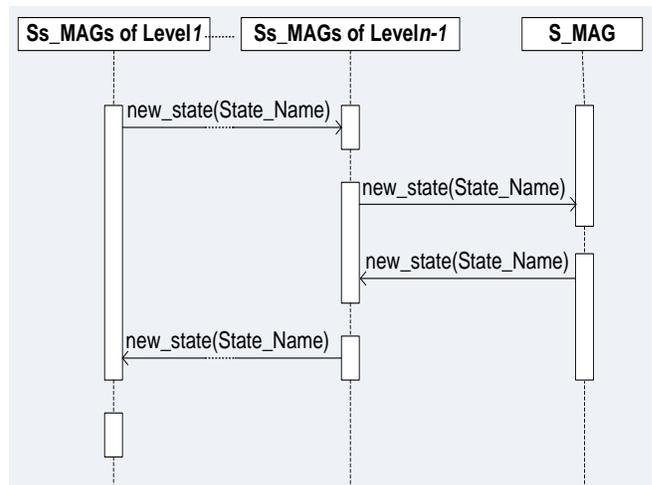


Figure 11. Collaboration protocol among agents across hierarchical levels.

According to the collaboration protocol of Fig. 11, every new state that results from a state transition at level l is communicated by the agent to its parent agent, which in turn communicates its own new state higher up to its parent, and so on successively to the S_MAG at the top level (level n). The S_MAG, in turn, communicates its own new state to the children at level $n-1$. Every child agent communicates its own new state similarly to its children. This scenario is repeated successively between every agent and its children until the agents of level l are reached.

According to Fig. 12, Ss_MAGs of level l share their sensory measurements (global measurements). Any Ss_MAG may ask for a measurement by sending an ask message to the intended Ss_MAG. The receiving Ss_MAG (asked Ss_MAG) should answer accordingly by sending a tell message.

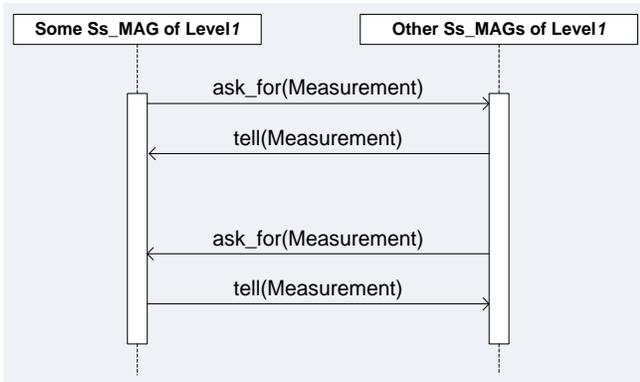


Figure 12. The collaboration protocol among MAGs of level l .

Every agent deployed at level l is provided with a portion of the monitoring model, which incorporates a state-transition table and a number of diagnostic models. An agent at this level is also provided with a monitoring algorithm to track the behaviour of the monitored sub-system and a diagnostic algorithm to relate the verified failure events to their underlying causes. Every agent deployed at levels extending from level 2 to level n is provided with a monitoring model, which is a state-transition of the assigned sub-system or system.

IV. CASE STUDY: AIRCRAFT FUEL SYSTEM

Fig. 13 shows a graphical illustration and components of the fuel system. The system functions to maintain safe storage and even distribution of fuel in two modes. The first is the consumption mode in which the system provides fuel to the port and starboard engines of a two-engine aircraft. The second is the refuel mode.

During the consumption mode and to maintain the central gravity and stability, a control unit applies a feedback-control algorithm to ensure even fuel consumption across the tanks; flow rates and directions are as shown in Fig. 13. Another algorithm is applied similarly to control the even distribution of fuel injected from the refuelling point to the tanks during the refuel mode. The system is arranged in four sub-systems: a central deposit (CD), left and right wing (LW, RW) deposits and an engine feed (EF) deposit, which connects fuel resources to the two engines. An active fault-tolerant control strategy is implemented; specifically, in the presence of faults there are alternative flow paths to connect the two engines to the available fuel resources.

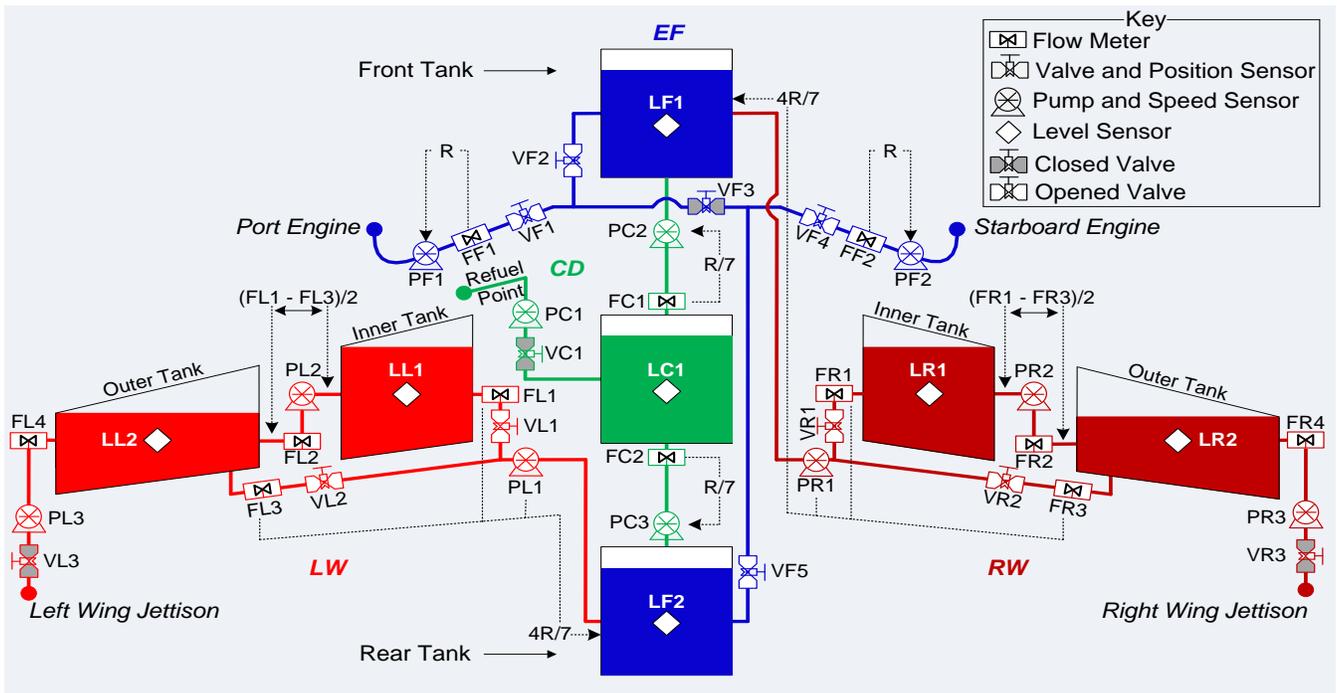


Figure 13. Graphical illustration of an aircraft fuel system.

As shown in Fig. 14, five monitoring agents are deployed to monitor the fuel system. Four of those agents are to monitor the four sub-systems; they appear as EF_MAG, CD_MAG, LW_MAG, and RW_MAG. The fifth agent is AFS_MAG, which monitors the entire fuel system. The monitor is implemented using Jason interpreter, which is an extended version of AgentSpeak programming language [72].

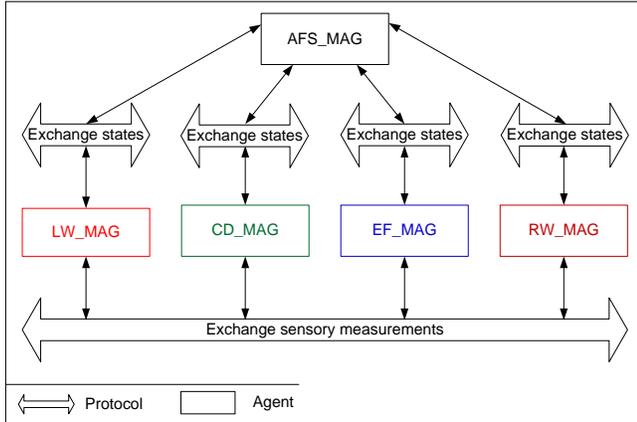


Figure 14. Deployment of agents to monitor the aircraft fuel system

To achieve the monitoring experiment, excerpts of the state-transition tables of the CD, LW and RW sub-systems are as shown by Table III, Table IV and Table V.

TABLE III. STATE-TRANSITION TABLE OF THE CD SUB-SYSTEM.

CURRENT STATE	CONDITIONS	EVENT	ALARM	CONTROLLING	DIAGNOSIS	NEW STATE
CD_CM_EFS1	Assessment: CD sub-system operates normally. Guidance: none	$(T(FC1 > R/7 + 0.03, 6) \text{ AND } T(FC1 > FC2 + 0.03, 6)) \text{ OR } (T(FC1 < R/7 - 0.03, 6) \text{ AND } T(FC1 < FC2 - 0.03, 6))$	CD sub-system has a sensory failure.	- FC1= FC2;	Sensor FC1 has failed.	CD_CM_PDS1
		AFS_CM_PDS1 == true;	none	- FC1 = -R/5; - FC2 = 3R/5;	not_needed	CD_CM_EFS2
		AFS_CM_PDS2 == true;	none	- FC1 = -3R/7; - FC2 = 4R/7;	not_needed	CD_CM_EFS3
CD_CM_PDS1	Assessment: CD sub-system operates degradedly. Guidance: none.	$VC1 == 1 \text{ AND } VF1 == 0 \text{ AND } VF4 == 0$	none	not_needed	not_needed	CD_RM_PDS1
CD_RM_PDS1	Assessment: CD sub-system operates degradedly. Guidance: flow meter FC1 must be replaced.	$T(FC2 < 0.03, 4)$	rear tank is not refueling.	- PC1 = 0; - VF1 = 0;	needed	CD_RM_FS
CD_CM_EFS3	Assessment: CD sub-system copes with a degraded state of the AFS. Guidance: none.	$T(FC1 < 0.03 \text{ OR } FC2 < 0.03, 4)$	abnormal flow from the central tank.	- PC2_S = 0; - PC3_S = 0;	needed	CD_CM_FS

TABLE IV. STATE-TRANSITION TABLE OF THE LW SUB-SYSTEM.

CURRENT STATE	CONDITIONS	EVENT	ALARM	CONTROLLING	DIAGNOSIS	NEW STATE
LW_CM_EFS1	Assessment: LW sub-system operates normally. Guidance: none	AFS_CM_PDS2 == true;	none	- FL3 = R/7; - FL1 = R/7;	not_needed	LW_CM_EFS2
		$V(LL1, 5) > I((FL1 + FL2, 5) + 0.06)$	inner tank of LW sub-system is leaky	- PL1 = 0; - VL1 = 0; - VL2 = 0; - VL3 = 1; - FL2 = -0.285; - FL4 = 0.571;	leak in the inner tank of LW.	LW_CM_TDS1
LW_CM_EFS2	Assessment: LW sub-system copes with a degraded state of the AFS. Guidance: none	$V(LL1_L, 5) > I((FL1 + FL2, 5) + 0.06)$	inner tank of LW sub-system is leaky	- PL1 = 0; - VL1 = 0; - VL2 = 0; - VL3 = 1; - FL2 = -0.285; - FL4 = 0.571;	leak in the inner tank of LW.	LW_CM_TDS1

TABLE V. STATE-TRANSITION TABLE OF THE RW SUB-SYSTEM.

CURRENT STATE	CONDITIONS	EVENT	ALARM	CONTROLLING	DIAGNOSIS	NEW STATE
RW_CM_EFS1	Assessment: RW sub-system operates normally. Guidance: none	AFS_CM_PDS1 == true;	none	- FR1 = 2R/5; - FR3 = 2R/5;	not_needed	RW_CM_EFS2
		AFS_CM_PDS2 == true;	none	- FR1 = R/7; - FR3 = R/7;	not_needed	RW_CM_EFS3
RW_CM_EFS2	Assessment: RW sub-system copes with isolating LW sub-system. Guidance: none	$V(LR1, 5) > I((FR1 + FR2, 5) + 0.06)$	inner tank of RW sub-system is leaky	- PR1 = 0; - VR1 = 0; - VR2 = 0; - VR3 = 1; - FR2 = -0.285; - FR4 = 0.571;	not_needed	RW_CM_TDS1
RW_CM_EFS3	Assessment: RW sub-system copes with a degraded state of the AFS. Guidance: none	$V(LR1, 5) > I((FR1 + FR2, 5) + 0.06)$	inner tank of RW sub-system is leaky	- PR1 = 0; - VR1 = 0; - VR2 = 0; - VR3 = 1; - FR2 = -0.285; - FR4 = 0.571;	not_needed	RW_CM_TDS1

Among the faults that have been injected to test the monitor is that the port engine is not fed and a fault of flow meter sensor FC1 of the central deposit (CD) sub-system.

A. First Simulated Failure scenario: "PEngine is not fed"

Once the monitoring agent EF_MAG evaluates expression (2) with true, it perceives the state-transition table (Table II) and achieves the following procedure:

- From the relevant ALARM attribute, agent FE_MAG quotes the statement "port engine is not fed" and alarms the pilot.
- From the relevant CONTROLLING attribute, agent FE_MAG checks the possibility of controlling that event. As the controlling depends on the underlying cause, that attribute accordingly tells the EF_MAG to achieve a diagnostic process by traversing the relevant fault tree ("after_diagnosis").
- From the relevant DIAGNOSIS attribute, agent FE_MAG verifies the need for a diagnostic process and updates the symptoms of the diagnostic model.
- From the relevant NEW STATE attribute, agent FE_MAG transits to the new state, which is the temporary degraded or failure state of the consumption mode EF_CM_TDFS1. From this state the pilot is provided with the assessment, "port engine is not fed and recovery is in progress" and guidance, "watch for further feedback".

- Agent EF_MAG also communicates the current state to the parent agent (AFS_MAG). The state does not trigger a state transition in the state-transition table of the AFS_MAG. At this point, the pilots are alarmed and informed on the operational condition as shown in Fig. 15.

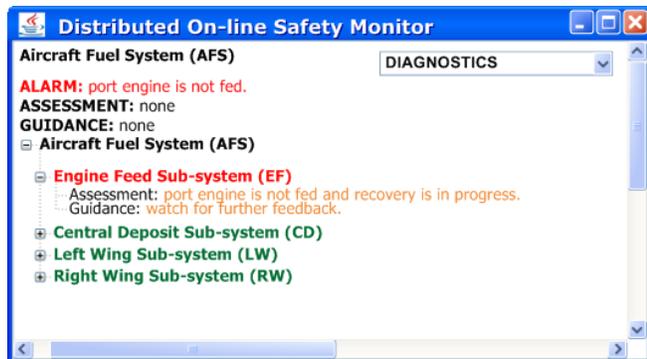


Figure 15. Operator interface after detecting and verifying the failure.

Since a diagnostic process is needed, agent EF_MAG retrieves the position of the top node of the relevant fault tree and launches a diagnostic process before launching a monitoring cycle for the new state EF_CM_TDFS1. By traversing the relevant diagnostic model (Fig. 8) the underlying cause is diagnosed and the required corrective measures are taken. Assuming that the cause is “VF1’s commission close command”, controlling is not possible and thus agent EF_MAG perceives Table II and achieves the following procedure:

- Launches a monitoring cycle to the active events of state EF_CM_TDFS1.
- During this cycle the occurrence of event $T(|FF1_F| < 0.03, 4)$ is verified consequently.
- From the relevant ALARM attribute, agent EF_MAG quotes and announces an alarm of “feeding port engine cannot be recovered”.
- From the relevant CONTROLLING attribute, agent EF_MAG takes the following actions: switching pump PF1 off and closing valves VF1 and VF2.
- As the diagnostic process appears not to be needed with this event, agent EF_MAG moves accordingly to the NEW STATE attribute, identifies and transits to a new state, which is EF_CM_PDS1. From this state the pilot is provided with assessment as “port engine is not fed” and guidance, “none”.
- Agent EF_MAG also communicates EF_CM_PDS1 to the parent agent (AFS_MAG).

Feeding only one engine (starboard engine) requires changing the operational structure of the entire fuel system to maintain an even level across the seven tanks. Accordingly, the above procedure is not enough to control the fault; controlling these conditions requires global collaboration among the remaining three sub-systems: LW, RW, and CD.

Once agent AFS_MAG receives a message conveying state EF_CM_PDS1, it perceives the state-transition table (Table I) and achieves the following procedure:

- While the current state is AFS_CM_EFS, the received state results in verifying the occurrence of EF_CM_PDS1 == true.
- From the relevant NEW STATE attribute, agent AFS_MAG transits to the new state, which is the permanent degraded state AFS_CM_PDS2. From this state the pilot is provided with assessment as “port engine cannot be fed, whereas starboard engine is feeding normally” and guidance, “none”.
- Agent AFS_MAG also communicates state AFS_CM_PDS2 to the child agents: CD_MAG, LW_MAG and LW_MAG.

Upon receiving messages conveying that state, each child agent achieves a certain fault controlling procedure to draw the corresponding flow rates and also transits to a new state. State transition and controlling procedures are as follows:

Agent CD_MAG perceives the state-transition table (Table III) and achieves the following procedure:

- While the current state is CD_CM_EFS1, the received state results in verifying the occurrence of AFC_CM_PDS2 == true.
- As the relevant ALARM attribute holds “none”, no alarm is thus announced.
- From the relevant CONTROLLING attribute, agent CD_MAG applies the following flow rates: $FC1 = -3R/7$ and $FC2 = 4R/7$.
- As the relevant DIAGNOSIS attribute holds “not_needed”, a diagnostic process is not launched.
- From the NEW STATE attribute, agent CD_MAG transits to the new state, which is another error-free state CD_CM_EFS3. From this state the pilot is provided with assessment, “CD sub-system copes with a degraded state of the AFS” and guidance, “none”.

LW_MAG perceives the state-transition table (Table IV) and achieves the following procedure:

- While the current state is LW_CM_EFS1, the received state results in verifying the occurrence of AFC_CM_PDS2 == true.
- As the relevant ALARM attribute holds “none”, no alarm is thus announced.
- From the relevant CONTROLLING attribute, agent LW_MAG applies the following flow rates; $FL3 = R/7$ and $FL1 = R/7$.
- As the relevant DIAGNOSIS attribute holds “not_needed”, a diagnostic process is not launched.
- From the relevant NEW STATE attribute, agent LW_MAG transits to the new state, which is another error-free state LW_CM_EFS2. From this state the pilot is provided with assessment, “LW sub-system copes with a degraded state of the AFS” and guidance “none”.

RW_MAG perceives the state-transition table (Table V) and achieves the following procedure:

- While the current state is RW_CM_EFS1, the received state results in verifying the occurrence of AFC_CM_PDS2 == true.
- As the relevant ALARM attribute holds “none”, no alarm is thus announced.
- From the relevant CONTROLLING attribute, agent RW_MAG achieves the following flow rates: FR1 = R/7 and FR3 = R/7.
- As the relevant DIAGNOSIS attribute holds “not_needed”, a diagnostic process is not launched.
- From the NEW STATE attribute, agent RW_MAG transits to the new state, which is another error-free state RW_CM_EFS3. From this state the pilot is provided with assessment, “RW sub-system copes with a degraded state of the AFS”, and guidance, “none”.

After achieving all the above procedures the operational structure of the fuel system appears different as fuel to feed the starboard engine only is drawn evenly from the seven tanks.

B. Second Simulated Failure scenario: Sensory Failure

Once the monitoring agent CD_MAG evaluates expression (3) with true, it perceives the state-transition table (Table III) and achieves the following procedure:

- From the relevant ALARM attribute, agent CD_MAG quotes and announces the alarm, “CD sub-system has a sensory failure”.
- From the relevant CONTROLLING attribute, agent CD_MAG instructs the fuel system control unit to ignore measurements delivered by flow meter FC1 and depend alternatively on those delivered by flow meter FC2.
- From the relevant DIAGNOSIS attribute, agent CD_MAG quotes “Sensor FC1 has failed” and announces it as the diagnosed underlying cause.
- From the relevant NEW STATE attribute, agent CD_MAG transits to the new state, which is the permanent degraded state CD_CM_PDS1. From this state the pilot is provided with assessment, “CD sub-system operates degradedly” and guidance, “none”.
- Agent CD_MAG also communicates the current state CD_CM_PDS1 to the parent (AFS_MAG).

When the agent AFS_MAG receives a message that conveys state CD_CM_PDS1, it perceives the state-transition table (Table I) and achieves the following procedure:

- While the current state is AFS_CM_EFS, the received state results in verifying the occurrence of CD_CM_PDS1 == true.
- From the relevant NEW STATE attribute, agent AFS_MAG transits to the new state, which is the permanent degraded state AFS_CM_PDS3. From this state the pilot is provided with assessment, “CD

sub-system delivers degraded functionality” and guidance, “none”.

- Agent AFS_MAG communicates state AFS_CM_PDS3 to the child agents: EF_MAG, LW_MAG and RW_MAG. As this state does not instantiate any active events of the children, no state transition is triggered and they do not take any action.

To demonstrate the ability of the monitor to deliver timely prognosis, let us assume that after controlling the fault, the aircraft has landed and during the pre-flying phase the refuelling mode is launched. This mode is triggered when the following expression is verified true:

$$VC1 == 1 \text{ AND } VF1 == 0 \text{ AND } VF4 == 0;$$

Then agent CD_MAG perceives the state-transition table (Table III) and achieves the following procedure:

- Executes the event on the table.
- As the ALARM attribute holds “none”, no alarm is thus announced.
- As the relevant CONTROLLING attribute holds “none”, no action is taken.
- As the relevant DIAGNOSIS attribute holds “not_needed”, then diagnosis is not launched.
- From the relevant NEW STATE attribute, agent CD_MAG transits to the permanent degraded state of the refuelling mode CD_RM_PDS1. From this state the pilot is provided with prognosis of assessment, “CD sub-system has a sensory failure” and guidance, “Flow meter FC1 must be replaced”.
- Agent CD_MAG also communicates the current state CD_RM_PDS1 to the parent agent (AFS_MAG).

When agent AFS_MAG receives a message conveying state CD_RM_PDS1, it perceives the state-transition table (Table I) and achieves the following procedure:

- While the current state is AFS_CM_PDS3, the received state results in verifying the occurrence of CD_RM_PDS1 == true.
- From the relevant NEW STATE attribute, agent AFS_MAG transits to the new state, which is the permanent degraded state of the refuelling mode AFS_RM_PDS3. From this state the pilot is provided with prognosis of assessment, “fuel system is not ready to achieve refuelling” and guidance, “none”.
- Agent AFS_MAG communicates state AFS_RM_PDS3 to the children: EF_MAG, LW_MAG and RW_MAG. As this state does not instantiate any active events of the children, no state transition will be triggered and they do not take any action.

This prognosis would appear on the operator interface as shown by Fig. 16. It can be seen how the monitor avoids overwhelming the pilot with extra alarm information and

provides timely prognosis according to the evolutionary behaviour of the fuel system.

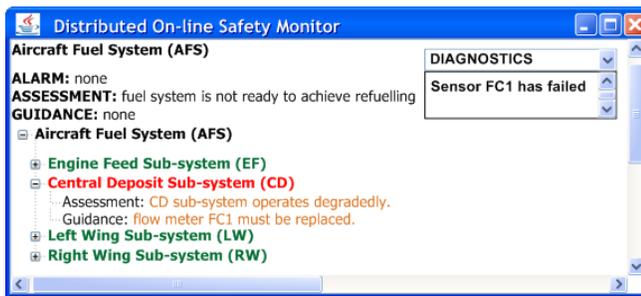


Figure 16. Operator interface provides the pilot with prognosis.

V. EVALUATION

The key aim of this work is to explore the concept of a novel architecture for a distributed safety monitor operating on a safety assessment model that hopefully could address some problems of earlier monitors and deliver effectively a broad range of safety monitoring tasks. Thus, it appears reasonable to compare the monitor developed herein with the earlier monitors and weigh their monitoring merits and drawbacks against each other.

In [12] and [13] two model-based monolithic monitors are developed to monitor critical applications and deliver fault detection and diagnosis, alarm annunciation and fault controlling. These monitors resemble the monitor developed herein in both the model-based approach and the deliverable monitoring tasks, but differ in their monolithic nature.

The monitoring model developed in [12] is quite similar in many ways to the monitoring model of the monitor developed herein; it too can be derived from the HiP-HOPS assessment model. The author of [12] has identified the following limitations:

- Vulnerability to sensor failures.
- The centralised nature of the architecture has limited the applicability of the monitor and made it unable to scale up to monitor large-scale and distributed systems (e.g., nuclear power plants or chemical processes).

The monitor developed herein addresses to some extent these limitations via the following:

- The exploitation of techniques to validate sensor measurements, to a certain extent. With careful use of functional and hardware redundancy, single sensor failures can be captured and masked.
- As it is based on a distributed concept in which monitoring agents are deployed according to the hierarchical architecture of the monitored system, the monitor has an extendable architecture that makes it able to scale up and monitor large scale systems.

The monolithic monitor developed in [13] is also:

- Vulnerable to sensor failures.
- Unable to scale up to monitor large-scale and distributed systems.

- Unable to provide the operators with prognosis

As mentioned above, the monitor developed herein is provided with the required techniques and provisions that address these limitations.

The data-based monolithic monitors of [29] and [73] are developed to detect and diagnose faults of chemical processes. These monitors have a narrower scope than that of the developed monitor and differ in their monolithic nature and data-based monitoring knowledge. In the case of [29] the monitor has been tested on a large number of components, but it has no provision to cope consistently with dynamic behaviour and does not deal with sensor failures, unlike the herein-developed monitor, which is able to track and follow states and modes and has provision for sensory failures.

The model-based multi-agent monitors developed in [74], [75] and [76] are intended to be applied to large-scale and distributed processes. They match the herein-developed monitor in the delivery of this task and the exploitation of the model-based approach and multi-agent system. These monitors differ from the developed monitor in scope as they only focus on fault detection and diagnosis and they do not deliver the alarm organisation and fault controlling tasks.

In [74] and [75] the authors suggest the following limitations in their work:

- The monitor does not incorporate local diagnostic models. It depends, rather, on global diagnostic decision taken among the monitoring agents. This does not work well when more than one agent has faulty monitored conditions and in such a case the delivery of erroneous diagnostics is quite possible.
- The monitor is vulnerable to sensor failures.
- The monitor lacks a protocol for effective collaboration among its monitoring agents. In the currently implemented protocol there is no direct communication among the agents and messages may take a long time to be transmitted from one agent to another until they reach the intended agents. This delay could result in ineffective monitoring.

The monitor developed herein addresses to some extent these limitations with the following provisions:

- Providing every monitoring agent of level l with a diagnostic algorithm and a number of diagnostic models, so they can achieve local diagnosis and deliver accurate diagnostics.
- Applying techniques to filter and validate sensor measurements and detect, diagnose and control single sensor failure.
- Developing collaboration protocols by which messages can be exchanged among the agents directly and with no delay.

Two limitations have been observed in the monitor developed in [76]. Both concern the diagnostic process and can be listed as follows:

- As the diagnosis is achieved globally and depends mainly on exchange of messages among the high level agent and lower level agents, this may place a

heavy communication load on the higher level agent and consequently result in its late response.

- As the diagnostic decision is processed globally and based on identifying the anomalies among the consistent conditions, the appearance of a number of anomalous conditions could potentially mislead the diagnostic process.

As a precaution against such limitations, the monitor developed herein has been provided with the following strategies:

- The communication load is reduced, as the monitoring agents of level *I* are provided with diagnostic models and algorithms so they achieve a local and independent diagnostic process.
- The diagnostic process is achieved based on local observations of every sub-system and it is not affected by anomalous conditions of other sub-systems.

The data-based multi-agent monitors in [61] and [62] are developed to detect and diagnose faults of dynamic chemical processes. They match the herein-developed monitor in the delivery of this task and exploitation of the multi-agent system and they differ in their data-based monitoring knowledge. The monitor of [62] can detect and diagnose both single and multiple faults. Practically, this is an outcome of exploiting sensor fusion methods and also global fusion collaboration among the agents. Similarly, the herein-developed monitor is able to detect, diagnose and moreover control single and multiple faults (but not multiple dependent sensor failures). This has been materialised by providing agents of level *I* with effectively formalised monitoring expressions and models to achieve local detection and diagnosis. Moreover, across the hierarchical levels, agents collaborate to achieve global reasoning over the entire monitored process.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a distributed on-line safety monitor based on a multi-agent system and knowledge derived from model-based safety assessment. Agents exploit that knowledge to deliver a range of safety monitoring tasks extending from fault detection and diagnosis to alarm annunciation and fault controlling. The delivery of these tasks has been discussed and demonstrated in the context of a study of an aircraft fuel system.

The monitor can detect symptoms of failure as violations of simple constraints, or deviations from more complex relationships among process parameters, and then diagnose the causes of such failures. With appropriate timed expressions, the monitor can filter normal transient behaviour and spurious measurements. Furthermore, the monitor is able to validate sensory measurements, detect, diagnose and control faulty sensors.

By exploiting knowledge about dynamic behaviour, the monitor can also determine the functional effects of low-level failures and provide a simplified and easier to

comprehend functional view of failure. Finally, by knowing the scope of a failure, the monitor can apply successive corrections at increasingly abstract levels in the hierarchy of a system.

Despite encouraging results, certain research issues remain to be investigated. The first is that the quality of the monitoring tasks and the correctness of the inferences drawn by the monitor depend mainly on the integrity and consistency of the monitoring model. The validation of the monitoring model is, therefore, an area for further research. Secondly, more work is needed on uncertainty of the diagnostic model and the application of the three-value logic. For that purpose, the incorporation of Bayesian Networks will be investigated in the future.

ACKNOWLEDGMENT

The author would like to thank Professor Yiannis Papadopoulos (University of Hull).

REFERENCES

- [1] A. Dheedan and Y. Papadopoulos. "Model-Based Distributed On-line Safety Monitoring," Proc. The Third International Conference on Emerging Network Intelligence (EMERGING 2011). Lisbon, Portugal, 20-25 November 2011.
- [2] C. Billings, "Human-centred Aircraft Automation: A Concept and Guidelines," Field CA United States: NASA Technical Memorandum TM-103885, NASA Ames Research Centre, Moffett, 1991. Available: http://www.archive.org/details/nasa_techdoc_19910022821 [Accessed 15th June 2012].
- [3] I. Kim, "Computer-based Diagnostic Monitoring to Enhance the Human-machine Interface of Complex Processes," Proc. Power Plant Dynamics, Control and Testing Symposium. Knoxville, TN, United States 27-29 May 1992.
- [4] V. Venkatasubramanian, R., Rengaswamy, K. Yin and S. Kavuri, "A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-based Methods," Computers and Chemical Engineering, 27(3), 2003, pp 293-311.
- [5] Y. Zhang and J. Jiang, "Bibliographical Review on Reconfigurable Fault-tolerant Control Systems," Annual Reviews in Control, 32(2), 2008, pp 229-252.
- [6] D. Pumfrey, "The Principled Design of Computer System Safety Analyses", DPhil Thesis, University of York, 1999.
- [7] J. Ma and J. Jiang, "Applications of Fault Detection and Diagnosis Methods in Nuclear Power Plants: A Review," Progress in Nuclear Energy, 53(3), 2011, pp 255-266.
- [8] I. Kim and M. Modarres, "Application of Goal Tree Success Tree Model as the Knowledge-base of Operator Advisory Systems," Nuclear Engineering and Design, 104 (1), 1987, pp 67-81.
- [9] M. Modarres and S. Cheon, "Function-centred Modelling of Engineering Systems Using the Goal Tree Success Tree Technique and Functional Primitives," Reliability Engineering & System Safety, 64(2), 1999, pp 181-200.
- [10] D. Chung, M. Modarres and R. Hunt, "GOTRES: an Expert System for Fault Detection and Analysis," Reliability Engineering & System Safety, 24(2), 1989, pp 113-137.
- [11] L. Felkel, R. Grumbach and E. Saedtler, "Treatment, Analysis and Presentation of Information about Component Faults and Plant Disturbances," Proc. symp Nuclear Power

- Plant Control Instrument, IAEA-SM-266/40, 1978, pp 340–347.
- [12] Y. Papadopoulos, “Model-Based System Monitoring and Diagnosis of Failures Using State-Charts and Fault Trees,” *Reliability Engineering and System Safety*, 8(3), 2003, pp 325-341.
- [13] H. Peng, W. Shang, H. Shi and W. Peng, “On-Line Monitoring and Diagnosis of Failures Using Control Charts and Fault Tree Analysis (FTA) Based on Digital Production Model.” *Proc. 2nd International Conference on Knowledge Science, Engineering and Management (KSEM’07)*. Lecture Notes in Computer Science (4798/2007). Berlin, Heidelberg: Springer, 2007, pp 544-549.
- [14] M. Maurya, R. Rengaswamy and V. Venkatasubramanian, “A Signed Directed Graph-based Systematic Framework for Steady-state Malfunction Diagnosis inside Control Loops,” *Chemical Engineering Science*, 61(6), 2006, pp 1790–1810.
- [15] G. Dong, W. Chongguang, Z. Beike and M. Xin, “Signed Directed Graph and Qualitative Trend Analysis Based Fault Diagnosis in Chemical Industry,” *Chinese Journal of Chemical Engineering*, 18(2), 2010, pp 265-276.
- [16] S. Narasimhan, P. Vachhani and R. Rengaswamy, “New Nonlinear Residual Feedback Observer for Fault Diagnosis in Nonlinear Systems,” *Automatica*, 44(9), 2008, pp 2222-2229.
- [17] A. Zolghadri, D. Henry and M. Monsion, “Design of Nonlinear Observers for Fault Diagnosis: a Case Study,” *Control Engineering Practice*, 4(11), 1999, pp 1535-1544.
- [18] T. Chen, and R. You, “A Novel Fault-Tolerant Sensor System for Sensor Drift Compensation,” *Sensors and Actuators A: Physical*, 147(2), 2008, pp 623-632.
- [19] T. El-Mezyani, D. Dustegor, S. Srivastava and D. Cartes, “Parity Space Approach for Enhanced Fault Detection and Intelligent Sensor Network Design in Power Systems,” *Proc. IEEE’2010 Conference on Power and Energy Society General Meeting*. Minneapolis, MN, 25-29 July 2010, pp 1-8.
- [20] M. Borner, H. Straky, T. Weispfenning and R. Isermann, “Model Based Fault Detection of Vehicle Suspension and Hydraulic Brake Systems,” *Mechatronics*, 12(8), 2002, pp 999-1010.
- [21] M. Abdelghani and M. Friswell, “A Parity Space Approach to Sensor Validation,” *proc. of the International Society for Optical Engineering (SPIE’2001)*. USA, Bellingham: Society of Photo-Optical Instrumentation Engineers, ISSN: 0277-786X CODEN, 4359 (1), 2001, pp 405-411.
- [22] W. Nelson, “REACTOR: an Expert System for Diagnosis and Treatment of Nuclear Reactor Accidents,” *proc AAAI* 82, August, 1982, pp 296-301.
- [23] T. Ramesh, S. Shum and J. Davis, “A Structured Framework for Efficient Problem-Solving in Diagnostic Expert Systems,” *Computers and Chemical Engineering*, 12(9-10), 1988, pp 891-902.
- [24] T. Ramesh, J. Davis and G. Schwenzler, “Catcracker: an Expert System for Process and Malfunction Diagnosis in Fluid Catalytic Cracking Units,” *proc. Annual Meeting of the American Institute of Chemical Engineering (AIChE)*, November 1989, San Francisco, CA.
- [25] S. Rich, V. Venkatasubramanian, M. Nasrallah and C. Matteo, “Development of a Diagnostic Expert System for a Whipped Toppings Process,” *Journal of Loss Prevention in the Process Industries* 2 (3), 1989, pp 145-154.
- [26] M. Maurya, R. Rengaswamy and V. Venkatasubramanian, “Fault Diagnosis by Qualitative Trend Analysis of the Principal Components,” *Chemical Engineering Research and Design*, 83 (9), 2005, pp 1122-1132.
- [27] M. Maurya, R. Rengaswamy and V. Venkatasubramanian, “A Signed Directed Graph and Qualitative Trend Analysis-Based Framework for Incipient Fault Diagnosis,” *Chemical Engineering Research and Design*, 85(10), 2007, pp 1407-1422.
- [28] M. R. Maurya, P. K. Paritosh, R. Rengaswamy and V. Venkatasubramanian, “A Framework for On-line Trend Extraction and Fault Diagnosis,” *Engineering Applications of Artificial Intelligence*, 23(6), 2010, pp 950-960.
- [29] L. A. Rusinov, I. V. Rudakova, O. A. Remizova and V. Kurkina, “Fault Diagnosis in Chemical Processes with Application of Hierarchical Neural Networks,” *Chemometrics and Intelligent Laboratory Systems*, 97 (1-15), 2009, pp 98-103.
- [30] N. Kaistha and B. Upadhyaya, “Incipient Fault Detection and Isolation of Field Devices in Nuclear Power Systems Using Principal Component Analysis,” *Nuclear Technology*, 136, 2001, pp 221-230.
- [31] J. Miller, “Statistical Signatures Used with Principal Component Analysis for Fault Detection and Isolation in a Continuous Reactor,” *Journal of Chemometrics*, 20(1-2), 2006, pp 34-42.
- [32] S. Wold, A. Ruhe, H. Wold and W. Dunn, “The Collinearity Problem in Linear Regression, the Partial Least Squares (PLS) Approach to Generalized Inverses,” *SIAM Journal of Science Statistical Computer*, 5(1984), 1984, pp 735-743.
- [33] S. Hwang, J. Lin, G. Liang, Y. Yau, T. Yenn, and C. Hsu, “Application Control Chart Concepts of Designing a Pre-alarm System in the Nuclear Power Plant Control Room,” *Nuclear Engineering and Design*, 238(12), 2008, pp 3522-3527.
- [34] G. Jang, D. Seong, J. Keum, H. Park, and Y. Kim, “The Design Characteristics of an Advanced Alarm System for SMART,” *Annals of Nuclear Energy*, 35(6), 2008, pp 1006-1015.
- [35] W. Brown, J. O’Hara and J. Higgins, “Advanced Alarm Systems: Revision of Guidance and its Technical Basis,” *US Nuclear Regulatory Commission*, Washington, DC (NUREG-6684), 2000. Available: http://www.bnl.gov/humanfactors/files/pdf/NUREG_CR-6684.pdf [Accessed 22nd February 2012].
- [36] B. Oulton, “Structured Programming Based on IEC SC 65 A, Using Alternative Programming Methodologies and Languages with Programmable Controllers,” *proc IEEE Conference on Electrical Engineering Problems in the Rubber and Plastic Industries*, IEEE no: 92CH3111-2, IEEE service centre, 31-14 April, Akron, OH, USA, 1992, pp 18-20.
- [37] A. Ghariani, A. Toguyeni and E. Craye, “A Functional Graph Approach for Alarm Filtering and Fault Recovery for Automated Production Systems,” *proc 6th International Workshop on Discrete Event Systems (WODES’02)*. 02-04 October 2002, Zaragoza, Spain, pp 289-294.
- [38] J. Lee, J. Kim, J. Park, I. Hwang and S. Lyu, “Computer-Based Alarm Processing and Presentation Methods in Nuclear Power Plants,” *proc World Academy of Science, Engineering and Technology*. Library of Congress, Electronic Journals Library, 65(2010), 2010, pp 594-598.
- [39] C. Yu and B. Su, “Eliminating False Alarms Caused by Fault Propagation in Signal Validation by Sub-grouping,” *Progress in Nuclear Energy*, 48(4), 2006, pp 371-379.
- [40] P. Baraldi, A. Cammi, F. Mangili and E. Zio, “An Ensemble Approach to Sensor Fault Detection and Signal Reconstruction for Nuclear System Control,” *Annals of Nuclear Energy*, 37(6), 2010, pp 778-790.
- [41] I. Kim, M. Modarres and R. Hunt, “A Model-based Approach to On-line Disturbance Management: The

- Models," *Reliability Engineering and System Safety*, 28(3), 1990, pp 265-305.
- [42] I. Kim, M. Modarres and R. Hunt, "A Model-based Approach to On-line Process Disturbance Management: the Models," USA: System Research Centre: University of Maryland, SRC TR 88-111, 1988. Available from: http://drum.lib.umd.edu/bitstream/1903/4837/1/TR_88-111.pdf [Accessed 4th May 2012].
- [43] R. Clark, "Instrument Fault Detection," *IEEE Transactions on Aerospace and Electronic Systems*, 14(3), 1978, pp 456-465.
- [44] J. O'Hara, W. Brown, P. Lewis and J. Persensky, "Human-System Interface Design Review Guidelines," Energy Sciences & Technology Department and Brookhaven National Laboratory, Upton, NY 11973-5000, 2002. Available: <http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0700/nureg700.pdf> [Accessed 22nd February 2012].
- [45] J. O'Hara, W. Brown, B. Halbert, G. Skraaning, J. Wachtel and J. Persensky, "The Use of Simulation in the Development of Human Factors Guidelines for Alarm Systems," *proc 1997 IEEE 6th Conference on Human Factors and Power Plants, Global Perspectives of Human Factors in Power Generation*. 08 - 13 Jun 1997, Orlando, FL, USA, pp 1807-1813.
- [46] E. Roth, and J. O'Hara, "Integrating Digital and Conventional Human-System Interfaces: Lessons Learned from a Control Room Modernization Program," Division of Systems Analysis and Regulatory Effectiveness Office of Nuclear Regulatory Research U.S. Nuclear Regulatory Commission, Washington, DC 20555-0001 NRC Job Code W6546, 2002. Available: <http://www.nrc.gov/reading-rm/doc-collections/nuregs/contract/cr6749/6749-021104.pdf> [Accessed 23rd February 2012].
- [47] J. Anderson, "Alarm Handler User's Guide," The University of Chicago, as Operator of Argonne National Laboratory, Deutsches Elektronen-Synchrotron in Der Helmholtz-Gemeinschaft (DESY) and Berliner Speicherring-Gesellschaft fuer Synchrotron-Strahlung mbH (BESSY), 2007. Available: <http://www.slac.stanford.edu/comp/unix/package/epics/extensions/alh/alhUserGuide.pdf> [Accessed 23rd May 2012].
- [48] J. Jiang, "Fault-tolerant Control Systems an Introductory Overview," *Automatica SINCA*, 31(1), 2005, pp 161-174.
- [49] R. Patton, "Fault-tolerant Control: the 1997 Situation," *proc 3rd IFAC symp on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'97)*. August 1997, Hull, United Kingdom, pp 1033-1055.
- [50] C. Seo and B. Kim, "Robust and Reliable H_{∞} Control for Linear Systems with Parameter Uncertainty and Actuator Failure," *Automatica*, 32(3), 1996, pp 465-467.
- [51] R. Srichander and B. Walker, "Stochastic Stability Analysis for Continuous-Time Fault Tolerant Control Systems," *International Journal of Control*, 57(2), 1993, pp 433-452.
- [52] I. Lopez and N. Sarigul-Klijn, "A Review of Uncertainty in Flight Vehicle Structural Damage Monitoring, Diagnosis and Control: Challenges and Opportunities," *Progress in Aerospace Sciences*, 46 (7), 2010, pp 247-273.
- [53] Delta Virtual Airlines, "Boeing 767-200/300ER/400ER Operating Manual," 3rd ed., Delta Virtual Airlines, 2003. Available: <http://www.deltava.org/library/B767%20Manual.pdf> [Accessed 5th March 2012].
- [54] Q. Zhao and J. Jiang, "Reliable State Feedback Control System Design against Actuator Failures," *Automatica*, 34(10), 1998, pp 1267-1272.
- [55] W. Heimerdinger and C. Weinstock, "A Conceptual Framework for System Fault Tolerance," Technical Report CMU/SEI-92-TR-033 ESC-TR-92-033, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 1992.
- [56] M. Shooman, "Reliability of Computer Systems and Networks: Fault Tolerance, Analysis and Design," USA, New York: John Wiley & Sons, Inc, 2002.
- [57] Y. Papadopoulos, J. McDermid, R. Sasse and G. Heiner, "Analysis and Synthesis of the Behaviour of Complex Programmable Electronic Systems in Conditions of Failure," *Reliability Engineering & System Safety*, 71 (3), 2001, pp 229-247.
- [58] X. Ren, H. Thompson and P. Fleming, "An Agent-based System for Distributed Fault Diagnosis," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 10(2006), 2006, pp 319-335.
- [59] S. Eo, T. Chang, B. Lee, D. Shin and E. Yoon, "Function-behaviour Modelling and Multi-agent Approach for Fault Diagnosis of Chemical Processes," *Computer Aided Chemical Engineering*, 9 (2001), 2001, pp 645-650.
- [60] S. Eo, T. Chang, B. Lee, D. Shin and E. Yoon, "Cooperative Problem Solving in Diagnostic Agents for Chemical Processes," *Computers & Chemical Engineering*, 24 (2-7), 2000, pp 729-734.
- [61] Y. Ng and R. Srinivasan, "Multi-agent Based Collaborative Fault Detection and Identification in Chemical Processes," *Engineering Applications of Artificial Intelligence*, 23(6), 2010, pp 934-949.
- [62] G. Niu, T. Han, B. Yang and A. Tan, "Multi-agent Decision Fusion for Motor Fault diagnosis," *Mechanical Systems and Signal Processing*, 21(3), 2007, pp 1285-1299.
- [63] C. Wallace, G. Jain and S. McArthur, "Multi-agent System for Nuclear Condition Monitoring," *proc of the 2nd International Workshop on Agent Technologies for Energy System (ATES'11)*, a workshop of the 10th International Conference of Agent and Multi-agent System (AAMAS'11), 2nd of May 2011, in Taipei, Taiwan.
- [64] A. Sayda, "Multi-agent Systems for Industrial Applications: Design, Development, and Challenges," In: Alkhatieb F., Al Maghayreh E., Abu Doush L., ed. *Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications*. Rijeka, Croatia: InTech, 2011, pp 469-494.
- [65] E. Mangina, "Intelligent Agent-based Monitoring Platform for Applications in Engineering," *International Journal of Computer Science & Applications*, 2(1), 2005, pp 38-48.
- [66] HSE, "Better Alarm Handling, HSE Information Sheet," UK: Health and Safety Executive, chemical sheet No 6, 2000. Available from: <http://www.hse.gov.uk/pubns/chis6.pdf>, [Accessed 13th June 2012].
- [67] E. J. Trimble, "Report on the Accident to Boeing 737-400 G-OBME near Kegworth, Leicestershire on 8 January 1989," Department of Transport Air Accidents Investigation Branch, Royal Aerospace Establishment. London, HMSO, 1990. Available from: http://www.aairb.gov.uk/cms_resources.cfm?file=/4-1990%20G-OBME.pdf [Accessed 5th April 2012]
- [68] BEA, "Safety Investigation into the Accident on 1 June 2009 to the Airbus A330-203, flight AF447," France: Bureau of Investigations and Analysis for the safety of civil aviation (BEA), 2011. Available: <http://www.bea.aero/fr/enquetes/vol.af.447/note29juillet2011.en.pdf> [Accessed 2nd March 2012].
- [69] M. Breen, "Experience of Using a Lightweight Formal Specification Method for a Commercial Embedded System Product Line," *Requirements Engineering Journal*, 10(2), 2005, pp 161-172.

- [70] S. McArthur, E. Davidson, J. Hossack, and J. McDonald, "Automating Power System Fault Diagnosis through Multi-agent System Technology," *proc 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, 5-8 Jan 2004, Big Island, Hawaii, pp 1-4.
- [71] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems: A Practical Guide*. Chichester: John Wiley & Sons LTD, 2004.
- [72] R. Bordini, J. Hubner, and M. Wooldridge, *Programming Multi-Agent Systems in AgentSpeak Using Jason*. UK, Chichester: Wiley, 2007.
- [73] X. Doan and R. Srinivasan, "Online monitoring of multi-phase batch processes using phase-based multivariate statistical process control," *Computers & Chemical Engineering*, 32(1-2), 2008, pp 230-243.
- [74] S. Y. Eo, T. S. Chang, B. Lee, D. Shin and E. S. Yoon, "Cooperative problem solving in diagnostic agents for chemical processes," *Computers & Chemical Engineering*, 24 (2-7), 2000, pp729-734.
- [75] S. Y. Eo, T. S. Chang, B. Lee, D. Shin and E. S. Yoon, "Function-behavior modeling and multi-agent approach for fault diagnosis of chemical processes," *Computer Aided Chemical Engineering*, 9 (2001), 2001, pp 645-650.
- [76] X. Ren, H. A. Thompson, and P. J. Fleming, "An agent-based system for distributed fault diagnosis," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 10(2006), 2006, pp 319-335.

Getting Anesthesia Online: The smartOR Network

Marcus Koeny
Marian Walter
Steffen Leonhardt

Julia Benzko
Klaus Radermacher

Michael Czaplík
Rolf Rossaint

*Chair for Medical Information Technology
RWTH Aachen University
Aachen, Germany
koeny@hia.rwth-aachen.de
walter@hia.rwth-aachen.de
leonhardt@hia.rwth-aachen.de*

*Chair of Medical Engineering
RWTH Aachen University
Aachen, Germany
benzko@hia.rwth-aachen.de
radermacher@hia.rwth-aachen.de*

*Department of Anesthesiology
University Hospital RWTH Aachen
Aachen, Germany
mzczaplík@ukaachen.de
rrossaint@ukaachen.de*

Abstract—In this article, the concept of a future manufacturer-independent network standard for operating rooms is presented. On the basis of this standard, the realization of an integrated networked operating room is discussed using two workstations: One for the surgical and one for the anesthesia workplace displaying and controlling further networked components like patient bench, endoscope or tracking system respectively. As the main focus of this work the anesthesia workstation with an integrated system for telesupervision is presented. The workstation consolidates information from patient monitor, anesthesia machine and syringe pumps. Furthermore, information from surgical devices are requested to support the anesthesiologist. As a medical application the integration of an extended alarm concept is discussed followed by an outlook on options such a system can offer in near future.

Keywords—telesupervision; anesthesiology; smart operating room; network; SOA; patient alarms

I. INTRODUCTION

Part of this work was previously published at the EMERGING 2011 in Lisbon [1]. The described materials and methods are part of a research project called smart operating room (smartOR). The project aims to develop a manufacturer independent standard for device connectivity in the operating room, the Open Surgical Communication Bus (OSCB). Therefore this article presents a motivation for the standard and a description of the state of the art and our vision. After that the protocol itself (OSCB) is described followed by the concepts.

A. Motivation

In Figure 1, a typical view of the anesthesiologist's workplace can be seen. Obviously there is a multitude of different medical devices like a patient monitor, an anesthesia machine, infusion pumps, operating room lights and surgical devices, like endoscopes and further instruments supporting the medical staff to treat the patient.

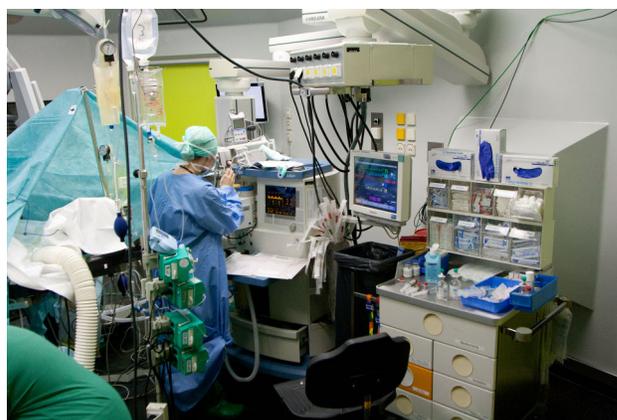


Figure 1. Typical scenario in an operating room [2]

All of these devices are necessary for a safe and successful process of the surgical intervention. However, the variety of devices, which are in most cases from different manufacturers, offer different risks for the patient and the operating staff. Each device has own operating panels and display concepts. Hence, information is presented to the user very individually and non-uniformly. For example, most infusion pumps have numeric displays with according numeric input. The patient monitor has a touch surface and the anesthesia machine is operated using a combination of setting valves and key input. Furthermore, in most cases there is a barrier between the anesthesiologist's workplace and the surgical workplace, due to hygienic reasons, which limits the interaction and information exchange between the anesthesia and surgical team. Nevertheless, despite of different tasks and focuses (performing surgery or controlling narcosis and patient's health state respectively), an interaction is required for optimal patient safety.

Currently, companies like for example Wolf [3] or Stryker [4], are building integrated operating room solutions present-

ing consolidated information from all devices in centralized displays. An example of such an integrated operating room can be seen in Figure 2. Consolidated displays can be



Figure 2. Operating room of the future. Example for consolidated data from various devices [3]

seen, which present optimized context-adapted information. Devices can be operated in a uniform way, so that usability concepts are clearly improved. Furthermore, several safety and assistance concepts can be integrated to support clinicians. Unfortunately, there are still drawbacks. Usually, integrated operating room systems are only built by one manufacturer or a limited range of partners. The operating rooms are optimized for only a certain type of interventions. Furthermore, these solutions are often based on proprietary protocols which do not allow a manufacturer-independent connection.

The aim of this manuscript is to enlighten these aspects in detail and to present our vision of the networked operating room of the future.

B. State of the Art

The situation presented in Figure 1 can be found in most hospitals, although the above mentioned integrated operating room solutions exist. A typical device set of the anesthesiologist's workplace consists of patient monitor, anesthesia machine, and syringe pumps. In still frequently used older equipment the only networked combination is the anesthesia machine and the patient monitor in order to display vital signs of the anesthesia machine at the patient monitor's display. Hence, the patient monitor only displays vital signs; the anesthesia machine provides a user interface to control the parameters for mechanical ventilation and anesthesia. New upcoming solutions for the anesthesiologist's workplace like the Draeger Smart Pilot View [5] [6] or the GE Healthcare Navigator Application Suite [7] are often expensive and, like the integrated operating room solutions, only connectable to a limited range of devices. The advanced integrated operating rooms present clearly arranged user machine interaction and support the staff during the intervention. The manufacturers of these integrated solutions are often developing devices for special purposes like endoscopy

or brain surgery. Therefore the application of these rooms is often limited to a specific field of interventions. The described integrated operating room solutions are based on proprietary protocols. Therefore only manufacturer with an according license are able to integrate their own device into these operating rooms. Especially, smaller manufacturers with innovative products are handicapped without a license. Furthermore, more and more hospitals would combine different devices from different manufacturers to improve patient's treatment with best cost efficiency. A manufacturer independent standard would make possible every combination of devices regardless which manufacturer has produced it and which intervention type is performed.

But why does such a manufacturer-independent standard does not exist yet [8]?

In medical applications every device must be certified to comply with national and international medical standards like the ISO 60601 [9] for electrical operating safety and according to medical device acts. Furthermore, a risk analysis according to the ISO 14971 [10] or the ISO 80001 [11] must be performed. In case of connected devices this certification and risk analysis must be done for every device combination. This procedure is associated with much effort and costs. In case of a manufacturer independent standard the effort is much bigger, because the multitude of devices is larger and the risks are much more complex to assess. Due to the multitude of devices available on the market there must be developed new methods for the safety concepts and the risk management. Furthermore, the interest by the bigger manufacturer is low establishing a manufacturer independent standard. However, the request by clinicians for such a manufacturer independent standards becomes bigger and bigger.

C. The Vision

Our vision is to develop a standard for manufacturer-independent inter-device communication based on a service-oriented architecture (SOA) and on well-known internet technologies. The Open Surgical Communication Bus (OSCB) is the implementation of the standard. The use of a SOA has been previously proofed by different projects like the orthoMIT [12], the FUSION [13] or the Medical Device Plug and Play (MDPnP) [14] projects. During the design time of the standard, aspects like risk management and certifiability play an important rule. Concepts to integrate risk management into the protocol and mechanisms to ensure the safety and reliability of the network will be integrated into the standard.

Based on the defined standard new concepts for device operation and display concepts will be developed. Furthermore, the patients safety should be increased and medical staff should be supported by intelligent alarm systems. First in the following Section the basic concept with most important components of the smart operating room (smartOR) is

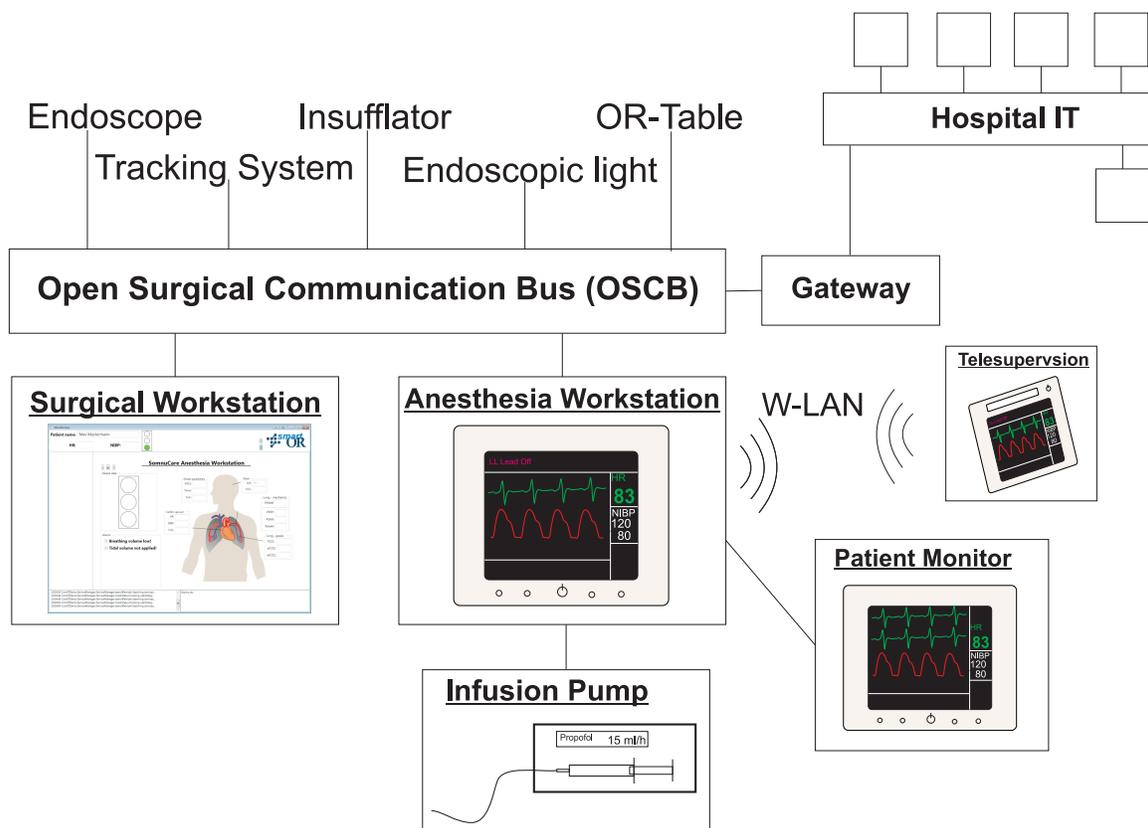


Figure 3. smartOR network overview modified after [1]

project participate in developing new standards and integrate their devices by using the already mentioned OSCB bus.

Figure 3 shows the anesthesia devices, which are currently not directly connected to the OSCB. A high amount of time critical data has to be transmitted from the device to the workstation, for example multiple waveforms like the electrocardiogram (ECG) or the photoplethysmogram (PPG). In most cases these data are not relevant for devices used by the surgeon in a non-consolidated form. Furthermore, at this time, these devices cannot be directly connected to the OSCB, because the software and interfaces on those device has to be adopted. Therefore, the anesthesia workstation is used as a protocol converter, which can forward relevant and requested data to the OSCB.

E. Gateway

For security and performance reasons, the OSCB of each operating room is an isolated network. Interaction between other ORs and the hospital IT network is realized by a gateway, which filters the traffic from the hospital IT. Additionally sensitive data from the OSCB is filtered. On the other hand patient data from the Hospital Information System or images from radiology (e.g. DICOM [16] format) can be transferred from the hospital IT through the gateway to the OR. Furthermore, updated patient records from

anesthesiology and surgery can be returned to the hospital information system.

F. Results

The above mentioned devices and methods are the basis for the smartOR network, which is realized using components and protocols described in Section III. In the smartOR the anesthesia and surgical workstations are the new display and control instruments and represent central components. The surgical and anesthesia devices can be controlled by the workstations. All components can be substituted by components from other manufacturer as long as these are using the OSCB standard.

III. TECHNICAL REALIZATION

In this Section, the technical principles of the OSCB are described. Since the standard is still in development, the current state of development is presented.

A. Internet Technologies Used

The communication layers are represented by the OSCB, which is described according to the ISO reference model in Figure 6. The physical and data link layer of the OSCB is based on the Ethernet standard IEEE802.3. Ethernet is a commonly used standard and has already been approved in many

medical applications like in the Medical Device Plug and Play MDPNP protocol [14][17] and Draeger Infinity network [18]. Moreover, Wi-Fi is used according to IEEE802.11 [19], to integrate mobile systems like the tablet PC for the telesupervision system. Finally the used physical layer does not matter, as long as used devices are conform to the ISO60601 [9] and according to medical device acts. Relevant for communication and for the protocol implementation are network and upper layers.

<u>Application</u>	HTTP, SOAP, DPWS
<u>Presentation</u>	
<u>Session</u>	
<u>Transport</u>	TCP, UDP
<u>Network</u>	IP (IPv4, IPv6)
<u>Data Link</u>	10/100/1000 Ethernet
<u>Physical</u>	Wireless LAN

Figure 6. Modified ISO / OSI reference model for the OSCB

The IP based TCP and UDP protocols are commonly used in the internet and local networks. Hence our network and transport layer is based on these protocols as well. The performance and transmitting characteristics (CSMA/CD) are sufficient as we have not to match real-time criteria with the OSCB. The OSCB is only defined for device identification, device management, exchange of parameters, alarms and commands with no hard real-time demands. Large amounts of data and real-time data, like high resolution video must be transmitted using other protocols. Additionally there can be other communication channels used, like a M2IO switch (comparable to a KVM switch with matrix functionality) or additional communication channels.

In the upper layers we use web service technology according to the Device Profile for Web Services (DPWS) [20] [21] standard. The DPWS standard combines SOAP [22] technology with functions for device discovery. Every device in the smartOR network implements methods for discovery, registration and deregistration. Therefore all devices in the operating room can be easily managed by a central management component, as a new device on the OSCB registers itself by the management component. Other devices can retrieve a list of all devices from the management component. If the central management component fails, devices can manage themselves using the integrated DPWS device

discovery functionality . The web services and exchanged messages are described in the following section in detail.

B. The Open Surgical Communication Bus

As described above, DPWS [20] [21] is used as the upper communication layers and builds the basis functionality in the OSCB. The OSCB defines the description of a DPWS device, web services hosted by a device and the messages exchanged by the web services.

Every device in the OSCB must implement three web services according to Figure 7:

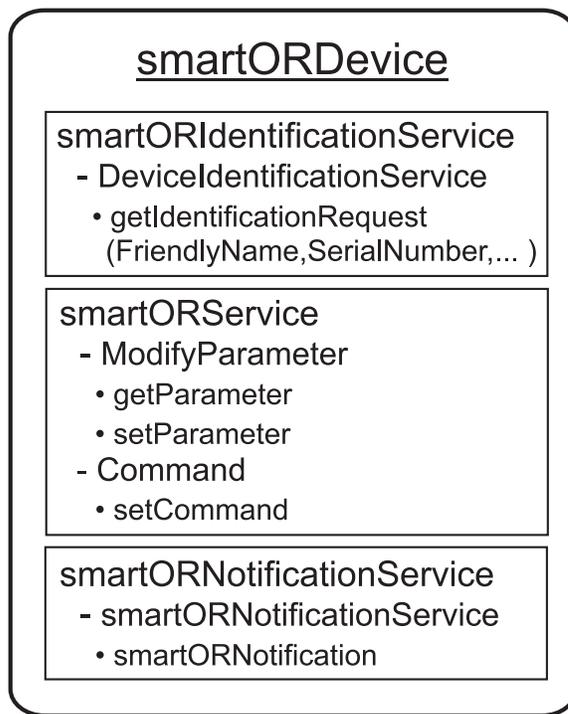


Figure 7. Web Services

- The *smartORIdentificationService* is used for device identification. It submits information about the device like hardware revision, software version and available parameters and commands.
- The *smartORWebService* is used for parameter exchange and command exchange. With the functions *getParameter* and *setParameter* various parameters like vital signs or device parameter can be read and set.
- The *smartORNotificationService* implements a notification service, which can be subscribed by clients. After subscribing the notification service every subscriber will be informed by the service about parameter changes or alarms.

Every device parameter, command and event has its own identification according to the ISO11073 [23]. The identifier is build using specified rules, which can be found in the standard. For example the designation of an insufflator device would be MDC_DEV_INSUFFLATOR. Where 'MDC' is the global identifier for parameters according to this standard and stands for medical device component. The second part specifies if it is a device (DEV), a command (CMD) or an event (EVT). Parameters are named using the device name as second part. For example 'MDC_ECG_HEART_RATE' means the heart rate derived from the ECG.

C. Results

Some parts of the OSCB are not specified yet; therefore even not all parts are realized in the software protocol. The usability of the SOA has been still proved [24] [25]. Currently just the WS4D-JMEDS framework in Java is used for the implementation of the OSCB. Therefore there is no comparison available between the WS4D-JMEDS framework and other framework's like WS4D-gSOAP or the .NET Micro framework DPWS implementation. Hence the WS4D-JMEDS framework is written in Java and offers much functionality a state of the art PC is necessary for implementation. The direct implementation in Java based medical devices is easy. Unfortunately, using this library in medical devices based on special embedded systems, still need effort or writing software interface for adaption.

IV. IMPLEMENTATION OF THE ANESTHESIA WORKSTATION

The anesthesia workstation is one of the central components of the smartOR and develops its full capabilities in combination with other components in the network. However, the workstation can be used standalone. For example the software has been tested in different studies connected only to a patient monitor, an anesthesia machine and syringe pumps. During the studies patient data were recorded during surgery in order to develop new alarm concepts, for example the one described in Section V.

The anesthesia workstation consists of a special medical PC, which is equipped with hardware interfaces like RS232 and Ethernet for medical device connection. Up to two screens can be used for visualization. The functionality is implemented in a software called SomnuCare, which is primary developed in C++ using the Qt [26] library. For easy development and integration of new components the software is separated into different modules.

The most important modules of the SomnuCare software can be seen in Figure 8. The central and most important component is the memory mapped file (MMF) engine. It stores and caches all data from the interfaces, where each MMF represents a single data stream from an interface. Such a data stream can be a waveform like an ECG, numeric vital signs like heart rate or alarms from a medical device. Data

acquisition is performed by the interfaces itself as described in Section IV-B. All interfaces in SomnuCare have a uniform API, which is described later in Section IV-B. This concept enables the programmer to add further interfaces for data acquisition without change of other parts of SomnuCare.

The telesupervision system is such a module and integral component of the SomnuCare software. This system consists of the anesthesia workstation as server component and a tablet-PC as remote component for the supervisor. The remote side is realized running an additional instance of the SomnuCare software, which is connected to the anesthesia workstation in the operating room using Wi-Fi.

The telesupervision server emulates an interface for the supervisors tablet PC and mirrors all data stored in the MMF engine to a special interface on the tablet PC.

A. Integration into smartOR Network

Currently only the Java based WS4D-JEMEDS framework is used for the OSCB protocol, as mentioned above. SomnuCare itself is written in C++. Therefore an additional module is used to integrate the web service functionality in SomnuCare. The program translates the web service requests to serialized strings, which are forwarded to the counterpart in SomnuCare using an IPC socket. The web service module makes use of the MMF engine to retrieve the requested data and sends them back to the Java program. Finally the Java program answers the request of the client.

B. Interfaces for Data Acquisition

As mentioned above the SomnuCare interfaces are used to read data from devices using different hardware interfaces and software protocols. For example a patient monitor can be connected via Ethernet to the workstation using an UDP multicast based protocol and an anesthesia machine using Draeger Medibus [27] protocol via an RS232 interface. The communication with the connected device is performed by the interface, so independently from the physical connection each device can be integrated in SomnuCare. The interface only has to implement the API functions shown in Figure 9 for configuring, starting and stopping the interface. Every interface holds an internal state machine, which is controlled by these functions. This enables SomnuCare to automatically handle different types of connected devices using a uniform API. The interfaces in SomnuCare are controlled by a manager module described in Section IV-D, which configures, starts and stops the interfaces and handles possible changes of state, for example caused by a disconnection. Received data is directly written by the interface to the memory mapped file engine.

C. Memory Mapped File Engine

The memory mapped file engine makes use of the operating system's memory mapped file API and supports reading and writing data to multiple segmented memory

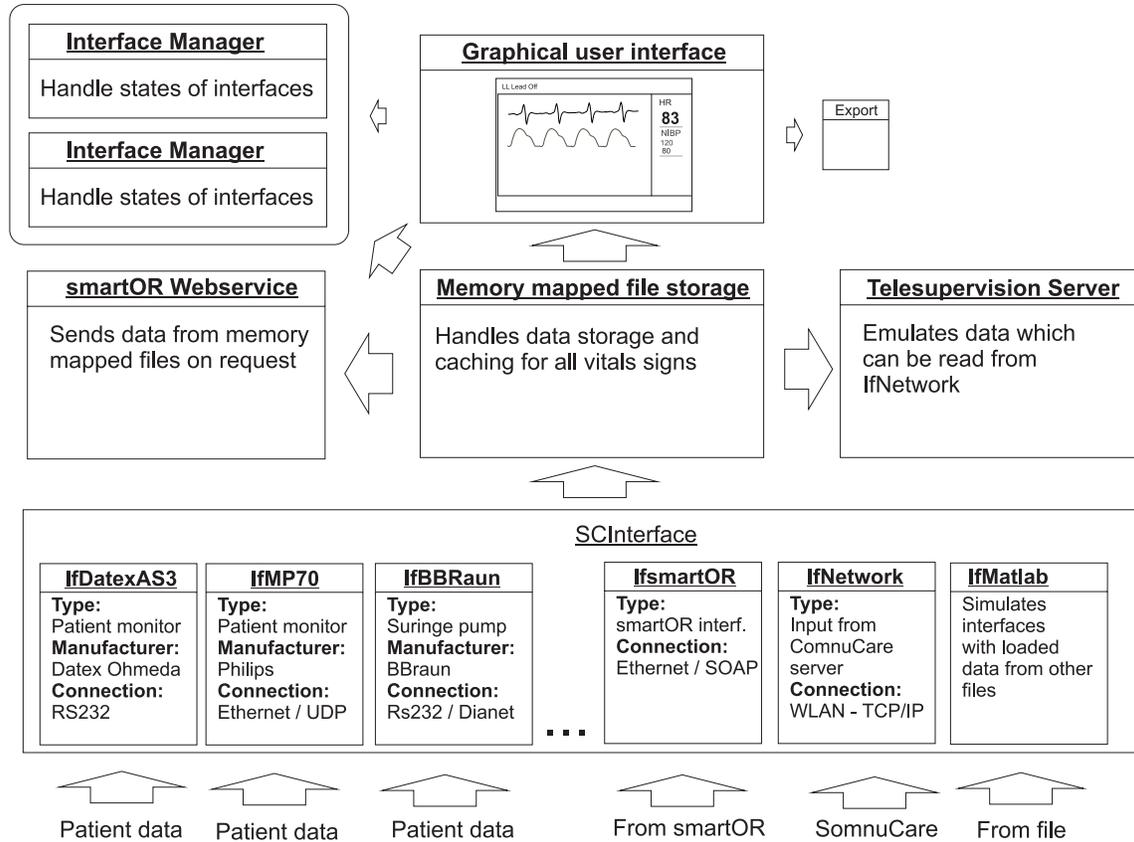


Figure 8. Structure of the SomnuCare software modified after [1]

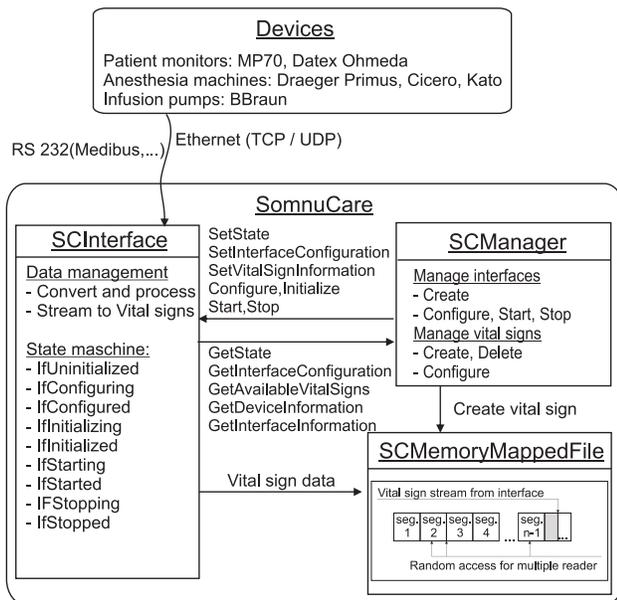


Figure 9. The SomnuCare interface

operating system functions. The memory mapped file technology allows data written to the memory mapped file being accessible like data in the RAM and stored to the hard disk like a log file at the same time. Hence only interfaces are used to write data to multiple memory mapped files, the engine supports only one writer for each memory mapped file, which can append data. Because all data should be logged and be available for other algorithms, functions for erasing or sorting data are not implemented. Reading is allowed by multiple instances as shown in Figure 10. This is necessary because the data from the interfaces is needed in different modules of the software. For example, the GUI as viewing element needs access to the last appended data, the telesupervision server must send the last incoming data, which have not already been send, to the client and the web services need to send specific data on request. During long interventions there can be a large amount of data stored in the MMF engine. For example during a 5 hour intervention with up to 50 recorded vital signs 500 MB data accumulate. To save memory not the whole file is mapped into memory. Therefore only the last segment is allocated. If the last segment is full, the file will be expanded with the segments size and this new segment will be allocated.

mapped files. Writing and reading data is performed by

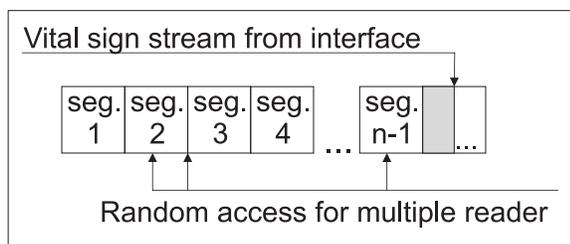


Figure 10. The SomnuCare interface

The reading function is able to randomly access segments. To improve the performance of multiple readers, they can register their instance for reading only appended data, which are tagged during insertion. In order to control all of these functions a special control segment is implemented in a separate memory mapped file. It stores information about segment handling and tagging new data for every reader instance. Furthermore, individual information like sample rate or data type can be stored in the control segment. This enables other programs to open the memory mapped file and use the contents for further evaluation.

Vital signs stored in the MMF engine and interfaces are controlled by a manager component, which is described in the following Section.

D. Manager Components

As mentioned above, the interfaces and vital signs are controlled by manager modules. The manager modules consists of hash lists, which can be stored to a database for persistence. During the runtime of SomnuCare the manager walks periodically through the lists in order to control the state and configuration of every interface or vital sign. Via the user interface the manager can be configured.

E. Telesupervision Server

The telesupervision server mirrors new appended data in all memory mapped files over the wireless network to the remote instance. Since the data in the memory mapped files is stored in a binary format, a binary protocol is used for data transmission. On the client side the counterpart is implemented as a standard SomnuCare interface. Therefore there is no modification needed on the client instance, except activating the according interface and setting the IP address of the server. Like all other interfaces the supervision interface must implement the state machine. This enables the network interface to automatically reconnect after a WLAN disconnect or any other failure. After a disconnect the interface tries to reconnect and load missing data in order to proceed with its normal operation. Due to the data tagging functionality and the consistency of the memory mapped files, no data will be lost, so an additional resynchronization is not necessary.

F. The Graphical User Interface

The graphical user interface of SomnuCare is designed to match the view of a standard patient monitor (Figure 11), because the patient monitor style has been proven for years. In order to test new features, alarm concepts and algorithms

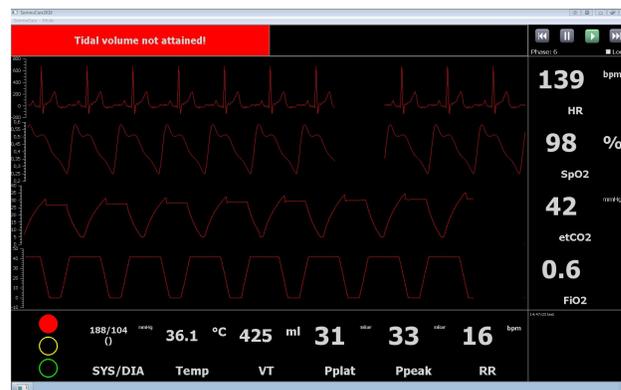


Figure 11. SomnuCare user interface [1]

the layout of the GUI can be modified. Furthermore, new elements can be inserted, for example a traffic light in order to visualize the state of a vital sign, as described later in Section V.

G. The Simulation Mode

Normally SomnuCare is used in an operating room scenario for interfacing other devices and supporting the anesthesiologist. Additionally SomnuCare can be used stand alone in studies with simulated or previously recorded data. Therefore a special interface (IfMatlab) is integrated, in order to load Matlab [28] workspaces and playback these data. Since IfMatlab acts as a standard interface and all data going through this interface are fed back into the MMF engine, allowing new algorithms to be evaluated.

H. Processors

A processor reads data from the memory mapped file engine and applies customized algorithms. The output of a SCProcessor is feed back into the memory mapped file system and therefore the computed output is available for different modules in SomnuCare.

For example a special SCProcessor instance is used to plot waveforms like ECG or real time blood pressure with a specific data rate. This is necessary because physicians are trained to make diagnosis based on standardized signals. Unfortunately, devices of different manufacturers transmit vital signs using different sample rates and block sizes. Therefore the processor must resample these signals and insert a time delay for uniform plotting.

I. Results

The SomnuCare software is a modular C++/Qt software, which is easily extensible with new features. It stores all data from connected devices in a memory mapped file engine and makes these data available for live and post-hoc processing. The workstation aims the look and feel of a patient monitor. However, it is not limited to the display of the patient monitor signals only, but is able to view data from patient monitor, anesthesia machine and syringe / infusion pumps. Furthermore, it is able to visualize new concepts like for example the display of the vital signs state.

SomnuCare itself has been used in different studies in the operating room recording data from patient monitors, anesthesia machines and syringe pumps from different manufacturers. The software will be permanently improved and extended in order to integrate new devices and try new concepts. The current realization of the OSCB integration is only a transitional solution.

The IPC communication between C++ and Java is afflicted with much overhead. Furthermore, receiving data from the OSCB using a web service client is not working. As long the SomnuCare software interfaces needed devices directly this is no drawback.

The performance of the Wi-Fi connection is sufficient to transmit all data processed in the workstation to the supervisor. However, there are still problems with the video/audio link between the operating room and the remote station. Especially transmitting patient data and establishing the audio link needs a good Wi-Fi infrastructure, because the bandwidth is not sufficient if the connection is weak.

V. MEDICAL APPLICATION

As mentioned in the introduction the vision of the smartOR network is to support the staff in the operating room. The presented smartOR concept and standard sets the base for the following presented medical application. In this Section a first approach to such a decision support and intelligent alarm system is described.

The data collected during the study described in the following Section are the basis for the presented alarm concepts. At the time the study took place, the SomnuCare software was not completed, therefore a special software module was developed to capture the data exclusively from the Datex Ohmeda AS/3 and BBraun perfusor serial interfaces and store them as comma separated text files. Relevant anesthesia and intervention-related events and milestones respectively were recorded by the same software. Thus, post-hoc synchronization was not required. Furthermore, the patients' sex, weight, age, size and the type of intervention were recorded.

A second study based on the SomnuCare software was carried out with a Philips MP70 patient monitor and BBraun syringe pumps. The setup was comparable to the described

study, which will be explained in the following Section more deeply.

A. Data Acquisition During Surgery

The data collected during the described studies are the base of the alarm system integrated in SomnuCare. The original study with a limited device combination has been previously published in [1].

In order to improve comparability of the collected data, similar surgical interventions, most of them gynecologic laparoscopic, were selected for recording. Anonymous data acquisition took place at the University Hospital Aachen after approval by the local ethics committee. Generally, the most important steps were pointed out as milestones:

- Start of presence of the anesthesiologist
- Start of anesthetization
- Approval for surgery
- Start of surgical preparation
- Start of surgical intervention
- End of surgical intervention
- End of surgical wrap-up
- End of anesthesia
- End of presence of the anesthesiologist

Furthermore, following events were recorded:

- Anesthetic events like intubation or inserting of a stomach tube
- Surgical events like skin incision, intra-operative relocation
- Intravenous drug injection

All vital signs and information from the following devices were recorded:

- Datex Ohmeda AS/3 or MP70 patient monitor
- Draeger Cicero, Cato, Primus anesthetic machine connected via Datex Ohmeda monitor
- Up to four BBraun perfusor infusion pumps

The above listed devices represent the standard setup in the University Hospital Aachen for these surgical interventions and resulted in the following recorded vital signs:

- Heart rate, non-invasive and/or invasive blood pressure, oxygen saturation
- Respiratory rate, tidal volumes, pressures, fractions of end-tidal CO₂, O₂ and anesthetic gases
- Anesthesia agents via syringe pumps and/or anesthetic gas concentration via the anesthetic machine

In total, data from 17 surgical interventions were recorded (8 female, 9 male patients). A balanced anesthesia with anesthetic gases (Isoflurane, Sevoflurane) was carried out in 8 cases; the remaining 9 received a total intravenous anesthesia using Propofol and Remifentanyl.

B. Anesthesia Alarm Concept

Generally, patient alarms are generated if a vital sign, for example the blood pressure, exceeds previously set limits.

In the operating room this concept is reasonable, since every exceeding can be critical. But, after critical situation, it is not necessary to repeat this alarm every 2 minutes, after the alarm has been confirmed and the underlying problem is already solved. As a solution for this problem, the following state machine was implemented for every important vital sign, for example like heart rate, non-invasive blood pressure and oxygen saturation: Compared to conventional alarms, which are triggered by exceeding pre-configured but fix limits, the state machine reconsiders the change of the vital sign after limit exceeding. So, initially the concept of classical limits is kept, but supplemented with the state machine algorithm.

Concretely, the alarm is rated into four conditions, similar to a traffic light.

- RED for a serious danger for the patient
- ORANGE for a situation with a potential danger for the patient
- YELLOW for the phase after a RED or ORANGE alarm is cleared
- GREEN for no alarm

The resulting state machine and flow diagram can be seen in 12.

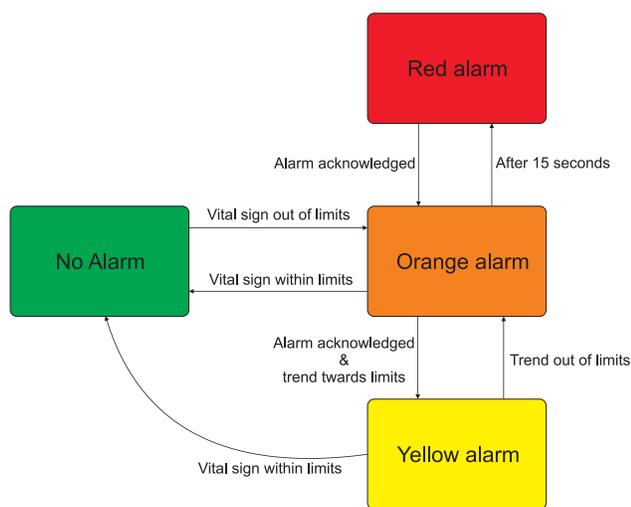


Figure 12. Resulting state maschine [1]

C. Results

The described traffic light alarm system is useful to avoid unnecessary alarms after critical situations. An example of such a situation can be seen in Figure 13. After a markedly increase of the blood pressure an orange alarm will be raised. If the alarm is acknowledged and the vital signs trend is targeting in-between the limits, the alarm turns into a yellow alarm. The yellow alarm still gives fair warning to the anesthesiologist, but the intrusive periodic acoustic warning is suppressed. If the trend is going worse, the alarm will return to the orange or red state.

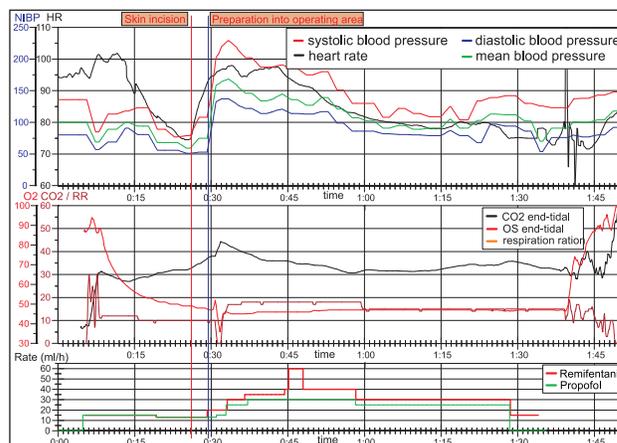


Figure 13. Example of an recorded surgical intervention [1]

VI. CONCLUSION AND FUTURE WORK

With regard to develop a manufacturer-independent protocol and interface standard for the operating room, first advances of the smartOR concept were described. In particular, the Open Surgical Communication Bus as well as approaches to realize workstations, which are custom-designed for anesthesia and surgery respectively were presented. First clinical data were already collected via already existing hardware interfaces and software modules and used for the development of novel anesthesia alarm concepts. Nevertheless the system is not fully developed yet but in progress.

A. smartOR System Overview

All devices in the smartOR are managed by a central device. Currently the device manager is integrated in the surgical workstation and lists all available devices in the user interface. However, the device manager should manage the access between the devices and manage the risk a new device or data exchange between devices can bring. Therefore additionally the device manager should integrate the risk and security management for the network.

Furthermore, fall back strategies must be planned in order to handle a failure of this component. Of course the network should be still functional even whether in a function limited mode. For example if the device manager fails, the network could keep its current state and ignore new devices, which risk and security aspects cannot be assessed by the manager.

The gateway is included in the concept, but not been implemented yet. It should consist of a powerful PC, which is able to run a stateful firewall, in order to filter unsecure traffic from the hospital IT, while forwarding relevant data, like DICOM images from the radiology or patient information from a hospital information system.

B. Technical Realization

The described components and protocols used in the lower layers of the OSI reference model, like Ethernet, TCP and UDP are used in many applications. Even in medical applications the used technologies are state of the art. Since there are no hard real time criteria to match, Ethernet with 100Mbit/s or 1Gbit/s is fast enough for the OSCB.

A comparison between different DPWS frameworks is still missing. The currently used WS4D-JEMEDS framework is a flexible tool, but not optimized for the OSCB. In future we will implement the OSCB protocol using different frameworks, like the in C++ written WS4D-gSOAP or the .NET Compact Framework, which covers the C# world. Additionally the consortium is developing an optimized framework in C++, with less dependencies to other libraries. This helps integrating the OSCB to medical devices, which are based on limited embedded hardware. The SomnuCare software will be modified to use the new C++ framework.

Another point is the cross operating room communication with the telesupervision client. As already mentioned the smartOR gateway filters all possible risky and unnecessary traffic and isolates each operating room. Implementing a cross operating room operation for the telesupervision system, the gateway must allow the traffic for the telesupervision system and forward it to the hospital IT or a WiFi infrastructure.

C. Implementation of the Anesthesia Workstation

The anesthesia workstation is a modular software, which integrates devices like patient monitor, anesthesia machine and syringe pumps into the OSCB. Furthermore, the software is capable of performing real time computations on the signals received from the interfaces. The concept has been approved in practical use in studies and in the demonstration environment, according to Figure 3.

One of the most important following steps is the direct integration of a DPWS framework, without external processes running, in order to reduce overhead and additional interface logic, which is not conform to the standard SomnuCare interface logic, described in Section IV-B. One possible solution would be the integration of the self-developed framework or the WS4D-gSOAP framework. Furthermore, work has to be done equipping syringe pumps and anesthesia machines with a control mode, so that parameters of connected devices can be changed by the workstation. This would help in situations where the anesthesiologist has to act near the patient and is not able to setup new dose rates or parameters on the devices. Realizing such functions needs an extension to the described concepts, mainly extensions in the risk management. The workstation must ensure the correct setting of dose. Furthermore, the user must be present in the operating room in order to supervise the new setup.

D. Medical Application

Several medical applications and advances can be achieved using the approach to reduce unnecessary but intrusive alarms is meaningful for practical and safety reasons. The conventional min/max alarms have been approved for many years and will therefore be kept and extended with technologies, which use the information from other devices in the smartOR network. For example the system can correlate possible blood pressure variability with the setting of the OR table. Than a workflow engine can analyze the actions of every device in the OR and determine possible future steps of the intervention in order to adopt alarm limits or to give advices for the next steps. Furthermore, dependent on the next steps of the intervention, the system can recommend, for example, to increase depth of anesthesia in advance of skin incision or a pending painful procedure.

ACKNOWLEDGEMENT

The described work is supported by the German Ministry of Economics and Technology. The smartOR project has the support code 01MMA09041A. The responsibility of this publication is by the authors. The authors would thank the following smartOR project partners for the efficient collaborative work:

- Innovation Center for Computer Assisted Surgery (IC-CAS) der Universitt Leipzig
- LOCALITE GmbH
- Richard Wolf GmbH
- SurgiTAIX AG
- Synagon GmbH
- VDE/DGBMT

Furthermore, the project is supported by clinics of different branches.

REFERENCES

- [1] M. Koeny, M. Czaplik, M. Walter, R. Rossaint, and S. Leonhardt, "A new telesupervision system integrated in an intelligent networked operating room," *The Third International Conference on Emerging Network Intelligence*, 2011.
- [2] M. Walter, "Telesupervision und automatisierung in der anaesthesie," *VDE Kongress 2006 Aachen*, 2006.
- [3] "Website Richard Wolf GmbH - Core system (last visited: December 2012) <http://www.richard-wolf.com/core.html>."
- [4] "Website Stryker Corporate (last visited: December 2012) <http://www.stryker.com>."
- [5] V. Billard, S. Ming, J. Miatello, K. Chatti, and F. Ferhi, "Drug interaction display during balanced anesthesia: Smart pilot view preliminary study," *Anesthesiology* 2010, no. A869, 2010.
- [6] R. Kennedy, "Seeing the future of anesthesia drug dosing: Moving the art of anesthesia from impressionism to realism," *Anesthesia & Analgesia*, vol. 111, no. 2, pp. 252-55, 2010.

- [7] "Website GE Healthcare (last visited: December 2012) <http://www.gehealthcare.com>."
- [8] B. Ibach, A. Zimolong, P. Knipp, J. Benzko, and K. Rademacher, "Modulare Integration und Vernetzung von Medizinprodukten unter Beruecksichtigung der IEC 80001," *DESIGN & ELEKTRONIK Entwicklerforum Embedded goes Medical*, 2009.
- [9] "Medizinische elektrische gerate," 2006.
- [10] "Application of risk management to medical devices," 2007.
- [11] "Application of risk management for it-networks incorporating medical devices," 2010.
- [12] "orthoMIT - Future Orthopedic Operating Room (Last visited: December 2012) <http://www.orthomit.de>."
- [13] "SOMIT - FUSION project (Last visited: December 2012) <http://http://www.somit-fusion.de>."
- [14] J. Goldman, "Advancing the adoption of medical device plug-and-play interoperability to improve patient safety and healthcare efficiency," Medical Device "Plug-and-Play" Interoperability Program, Tech. Rep., 2000.
- [15] J. B. nad Bastian Ibach and K. Rademacher, "Der traum vom plug u. play im op," *MED engineering*, vol. 03-04, pp. 76–79, 2011.
- [16] D. Clunie, *DICOM Structured Reporting*. Bangor: PixelMed Publishing, 2001.
- [17] "Medical device "plug-and-play" website (last visited: December 2012) <http://mdpnp.org>."
- [18] (2012, Jun.) Draeger infinity protocol. [Online]. Available: <http://www.draeger.de>
- [19] "IEEE 802.11 - Wireless LAN (WLAN), Wi-Fi."
- [20] "Devices Profile for Web Services Version 1.1," 2009.
- [21] T. Nixon and A. Regnier, "Devices profile for web services version 1.1," OASIS, Tech. Rep., 2009.
- [22] "W3C SOAP 1.2 - Simple Object Access Protocol," 2007.
- [23] "ISO 11073 - Health informatics," 2006.
- [24] B. Ibach, A. Zimolong, F. Portheine, and K. Rademacher, "Serviceorientierte Architekturen am Beispiel Operationssaal," 9. *Wuerzburger Medizintechnikkongress Kongressband*, pp. 198–201, 2008.
- [25] J. Benzko, B. Ibach, and K. Rademacher, "Or-integration based on soa - automatic detection of new service providers using dpws," *Proceedings CARS (Suppl. 1)*, vol. 01, pp. 195–196, 2010.
- [26] "QT website / layout management (last visited: December 2011) <http://qt.nokia.com/products/>."
- [27] *Draeger Medibus fuer Narkosegeraete*, 1999.
- [28] "The Mathworks company website (last visited: December 2012) <http://www.mathworks.com>."

Hierarchical Routing for Small World Wireless Networks

Juhani Latvakoski

VTT Technical Research Centre of Finland
Oulu, Finland
Juhani.Latvakoski@vtt.fi

Abstract — The number of embedded systems capable for wireless machine-to-machine service communication has continuously been increasing in recent years. In these kinds of dynamic ecosystems, the problems related to complexity and heterogeneity seriously challenges interoperability. As a contribution to this research, the small world paradigm from social sciences is being applied in a wireless networks context. A novel hierarchical networking concept, related routing algorithm and network optimization solutions are created to enable solving these problems. Logical short cuts are established between neighboring overlay nodes in order to avoid global flooding in distant route searches. In addition, physical short cuts may be created to remove the bottlenecks from the communication paths. The concept has been evaluated by graph theoretical analysis of the Hi-Search algorithm, simulation of the network optimization step and service discovery procedure. The evaluation results indicate that the algorithm with network optimization functions is able to lower the search delays, make the physical routes shorter and also improve throughput. In addition, solving the complexity and heterogeneity problems is made possible by localizing route search and abstracting communication to two hierarchical routing layers.

Keywords- *dynamic wireless networks; small world; routing*

I. INTRODUCTION

The number of wirelessly communicating embedded systems has been increasing continuously in recent years. This trend is assumed to lead to novel types of dynamic wireless networks, which are more and more necessary for communication between machines instead of only human-machine communication. Such dynamic wireless networks have previously been studied for example in the context of ad hoc and peer-to-peer overlay networks.

Ad hoc networks usually refer to a wireless network that can be established without any preceding configuration on the fly whenever required. The challenges in ad hoc networking solutions arise from the heterogeneity of operating environments, because of the different delay requirements, reaction times for route changes, power capabilities of the routing devices, and the limitations of the bandwidth usage, quality of service level and security. Because of these challenges, it can be assumed that the solution should be modular enough to enable smooth configuration and usage of multiple ad hoc routing protocols

in different domains of the network. When multiple ad hoc routing solutions are applied, then their interoperability will become one of the most critical requirements.

A well-known solution for solving the interoperability problem has been building overlay networks. In such an overlay network, a number of peers are connected to each other in a logical sense, and they can thus route messages between each other at a logical level even if no direct physical connections exist. Such solutions are able to improve robustness, availability, error resilience and even help in the transition to improved technological systems. One essential drawback of overlay networks is the overhead caused by the additional headers in the messages. Therefore, more processing power and memory is required in the overlay network nodes. However, there are still several open problems in communication between the nodes in dynamic wireless networks, such as heterogeneity of nodes, their dynamic existence, mobility, security, multiple radios, unreliable paths and topology, and continuous changes occurring in the network.

The motivation for the hierarchical routing arises from these challenges, especially complexity and heterogeneity of dynamic wireless networks. In addition, the wireless paths between communicating nodes usually tend to be too long and they go via nodes, which are not appropriate or willing to act as a router, which also makes the performance to be weak. Therefore, we focus here on hierarchical routing. This article is an extended version of the CTRQ 2012 conference paper [1]. The original CTRQ 2012 paper is here extended to clarify the main results of the hierarchical routing as a whole, including enhanced clarifications of the selected essential details also discussed in previous publications [2], [3], [4], and [5].

The selected approach for solving the problem in this research is the application of the small world paradigm for wireless networks. The small world paradigm has initially been studied in the context of social networks, where a small-world phenomenon has been detected [6], [7]. According to this, the average number of intermediate steps in a successful social communication chain is between five and six, “six degree of separation”. It is here expected that the well-connected nodes in wireless networks tend to behave in a networking sense like the well-connected people in social networks. Thus the small world paradigm from social sciences is here applied in wireless networks context. Based on this paradigm, a novel hierarchical networking

concept related routing and network optimization solutions and their evaluation results are provided in this work. The hierarchical route search algorithm provided is based on a graph theoretical system model and network search tree analysis both on overlay and at a physical level. Logical short cuts are established between neighboring overlay nodes to avoid global flooding in distant route searches. In addition, physical short cuts may be created in a network optimization step to make the end-to-end delays shorter, physical routes shorter and improve throughput. The Hi-Search algorithm is evaluated in terms of search path depths, number of control messages, and delay in the search, which are compared against the flat physical routing approach. The related network optimization step and procedures enabling service discovery for a user are evaluated by means of simulations.

The paper is organized as follows: The related work of small world wireless networks is described in section II. The conceptual system model for hierarchical networking and its reasoning is clarified in section III. The hierarchical routing solution is described in section IV. The simulation-based evaluation results are provided in section V, and finally, conclusions are given in section VI.

II. SMALL WORLD WIRELESS NETWORKS

A. *Small World*

The small world phenomenon originates from the observation that individuals are often linked by a short chain of acquaintances - "six degree of separation" [6], and [7]. Watts & Strogatz produced the network model, showing that rewiring a few links, called short cuts, in a regular graph can decrease the average path length between any two nodes while still maintaining a high degree of clustering between neighboring nodes [8]. The concept of small worlds is characterized by the facts that average path length is short and clustering is high degree. This means that most nodes are on average a few hops away from each other. High clustering means that most of the nodes' neighbors are also neighbors of each other. The small world phenomenon has been detected, for example, in email delivery experiments, and in the context of the Internet and the World Wide Web [9], [10], and [11].

Complex dynamic self-organizing wireless networks tend also to be scale-free [12]. They usually expand continuously by the addition of new nodes, and the new nodes tend to attach to nodes that are already well connected. The dynamic growth and preferential attachment lead to a scale-free property. Scale-free means that majority of nodes have very few neighbors, and only a few nodes have many neighbors. Thus, only a few well-connected nodes nicely connect a large number of poorly connected nodes. This phenomenon is independent of the network size, and such a scale-free network is also a small world.

Application of small world and scale-free features has also been studied in the context of wireless networks [13]. The dynamic wireless networks are spatial graphs that are usually much more clustered and have higher path length characteristics than random networks. In such a network, the

links depend on the radio range, which is usually a function of the distance. Adding a few wired short-cuts into the wireless networks, the degree of separation may be reduced drastically. Such short-cut links need not be random but may be confined to a limited number of hops, which is only a part of the network diameter.

Strategies for adding long-ranged links to centrally placed gateway node in wireless mesh networks are provided in [14]. The constraints of wireless networks, such as transmission range of long-ranged links (LL), limited radios per mesh router and limited bandwidth for wireless links are discussed. As a result, the constrained Small World Architecture for Wireless Mesh Networks is provided with three addition strategies of LL, which are able to provide a 43% reduction in average path length (APL). The LL addition strategies are random LL addition strategy (RAS), Gateway aware LL addition strategy (GAS), and Gateway aware greedy LL addition strategy (GAGS). In RAS, the links are randomly chosen and then some checks related to distance and the availability of radio are carried out. In GAS, there is an additional check and logic related to improving the gateway APL (G-APL). In GAGS, the logic for improving the G-APL is further optimized. Significant performance improvements in wireless mesh networks have been detected as the results of the LL addition strategies provided.

Summarizing, it has been discovered in the earlier theoretical small world-related research that, by adding a few short-cut links, average path length can be reduced significantly. However, the previous work has been mainly related to the application of wired links as short-cuts [13] or long-ranged links in mesh networks [14]. While in our approach, the dynamic wireless networking situation with multiple radio accesses, interoperability of routing protocols, and the problem related to the heterogeneity of nodes and links are taken as the starting point. Moreover, both logical and physical short cuts are created to solve these problems in practical situations.

B. *Routing Protocols*

The ad hoc networking protocols, such as, e.g., Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [15], Ad hoc On-Demand Distance Vector (AODV) [16], Optimized Link State Routing Protocol (OLSR) [17], and Dynamic MANET On-demand (DYMO) [18] are not optimal for specific operating environments due to differences in delay requirements, reaction times for route changes, the power capabilities of the routing devices, and the limitations of the bandwidth usage, quality of service level and security. Delay-Tolerant Networks (DTN) and opportunistic networking [19], [20], and [21] solutions enable communication also when the source and destination nodes are not necessarily reachable at the time of communication need. Therefore, usage of multiple ad hoc routing protocols optimized for different domains and situations of the dynamic wireless network should be enabled. A possible solution approach to these challenges is overlay networking. However, the *heterogeneity* of nodes,

radio links, and dynamic topologies still triggers challenges for both overlay networking and ad hoc networking systems.

There are multiple routing solutions implemented for overlay networking, such as the concept of a Content Addressable Network (CAN), which is a distributed application-level overlay infrastructure providing hash table functionality at an Internet-like scale [22]. A hash table is a data structure that efficiently maps keys into values. The CAN resembles a hash table, and the basic operations are the insertion, lookup and deletion of (key, values) pairs. Each CAN node stores a chunk (zone) of the entire hash table. In addition, the node holds information on a smaller number of adjacent zones in the table. Requests (insert, lookup, delete) for a particular key are routed by intermediate CAN nodes towards the CAN node whose zone contains that key. There are also several other routing overlay solutions, such as Chord [23], Tapestry [24], and Pastry [25]. Tapestry and Pastry differ from CAN and Chord in the sense that they take the network distances into account when constructing the routing overlay. SkipNet differs from Chord, CAN, Pastry and Tapestry in the sense that it provides controlled data placement and guaranteed routing locality by organizing data primarily by string names [26]. Tapestry, Chord, Pastry and CAN assume that most nodes in the system are uniform in resources such as network bandwidth and storage. Brocade provides a secondary overlay that exploits knowledge of the underlying network characteristics [27]. Usually, in peer-to-peer systems, nodes are connected to a small set of random neighboring nodes, and queries are propagated along these connections. Such a query tends to be very expensive in terms of bandwidth usage. A possible solution is the semantic overlay network (SON), which connects nodes having the same type of content to each other [28]. Queries are routed to the appropriate SONs, increasing the chances that matching files will be found quickly and reducing the search load on the nodes that do not have any related content. The hierarchical routing schemes with distributed hash tables (DHT) are discussed in [8]. The challenge with the DHT-based hierarchical routing schemes and also with most of the other overlay routing solutions is that they do not take physical level routing into consideration at all.

Small world-based routing, called SWER, dedicated to supporting sink mobility and small transfers has been provided in [29]. The hierarchy is based on clustering and cluster heads, and short cuts are applied for long-range links between clusters. The cluster head selects a sensor node to act as agent node to form the short-cut. The challenge in this solution is that the weak sensor nodes and radio links are still applied in realizing the short-cut. Hierarchical routing based on clustering using adaptive routing using clusters (ARC) protocol is provided in [30]. A new algorithm for cluster leader revocation to eliminate the ripple effect caused by leadership changes is provided. The ARC starts from the need to select a cluster leader. However, in our work we assume that the capability to act as a cluster head is preconfigured into the overlay nodes. Then there is no need to select a cluster head, but instead they need only to discover each other.

Helmy *et al.* have developed a contact-based architecture for resource discovery in large-scale wireless ad hoc networks (CARD) [31]. The mechanism is suitable for resource discovery as well as routing very small data transfers or transactions, in which the cost of data transfers is much smaller than the cost of route discovery. In CARD, resources within the vicinity of a node, up to a limited number of hops, are discovered using a proactive scheme. For resources beyond the vicinity, each node maintains links to a few distant nodes called contacts. The contacts help in creating an efficient way to query for distant resources. Two protocols for contact selection were introduced and evaluated: (a) probabilistic method, and (b) edge method, which was found to be a more efficient way for contact selection. Comparison with other schemes shows overhead savings reaching over 93% (vs. flooding) and 80% (vs. border casting or zone routing) for high query rates in large-scale wireless networks. The concept of contacts can be compared to our concept of overlay nodes. However, the contact nodes act as short-cuts in CARD, while our short-cuts are either logical or physical wireless links. Our approach in particular further enhances the system in such a way that the network optimization checks whether it is also possible to establish the physical wireless short cuts between overlay nodes as direct radio connections.

Variable-length short-cuts are constructed dynamically using mobile router nodes called data mules in disconnected wireless networks [32]. The data mules transfer data between nodes, which do not have a direct wireless communication link and belong to otherwise isolated networks. Their simulations indicate that even a small number of data mules can significantly reduce average path length. The overlay nodes might also act as mobile routers, but network optimization may not be possible or at least is not trivial in disconnected networks.

P2P network can be established using small world concepts, and it has been realized as SWOP, small world overlay protocol [33]. The average hop distance between P2P nodes can reduce the numbers of link traversals in object lookup, reduce the latency and can effectively satisfy a large number of users requesting a popular data object. However, the physical level routing is not taken into concern at all in the SWOP approach.

There are also quite a number of solutions for neighbor discovery such as [34], and [35]. However, route discovery is usually executed in a flat manner, e.g. [17]. The problem in such a search is that the search queries are also forwarded into the deep leaves of the search trees. Our approach is different in the sense that only the nearest logical overlay nodes are searched at the physical route level, and the network can be optimized by removing non-optimal radio links and physical routers from the path.

III. HIERARCHICAL NETWORKING CONCEPT

The applied system model of heterogeneous wireless network is shown in Figure 1. The system consists of heterogeneous nodes, which are shown using color codes for the different node types. In addition, the colors in the dotted circles represent the usage of different radio access types.

Each node may have one or more radio access capabilities, which can also be applied to temporarily connect the heterogeneous wireless network with legacy static Internet (blue clouds). The referred nodes may be switched on and off at any given time, which means that their presence is dynamic. Therefore, dynamic life cycle management is required for both the nodes and the networks.

The network nodes are categorized according to their capabilities. *U* node is a user interface (UI) node, which is able to host the network and services, which it may visualize for a user. *S* node is service node, which may provide set of services, act as super peer (cluster head) for services and overlay router. *R* node is a physical router node, which can route data traffic between different interfaces of the node. *T* node can for example be a sensor (*T_s*), actuator (*T_a*) or camera (*T_c*). *P* node is a special node in the sense that it is usually plugged in to be a logical part of *U* or *S* node. Each of the referred nodes may not always be on, and they may be mobile and can apply whatever wireless/wired access means for communication with the neighbor nodes.

The problem is related to the heterogeneity of nodes; some of the nodes do not have good capabilities for routing, while others do. For example, the radio access may not be power-efficient enough and the device may be battery-operated. In addition, some of them do not want to route at all for some owner-originated reasons. Having flat routing in such a system seems to lead to long path lengths and low performance, or even to the impossibility of establishing a connection at all.

The heterogeneous nodes may have several different radio accesses for communication with neighbor nodes. Some of the nodes may act as a relay for the specific radio technology. The lowest level of routing can thus be considered to be radio specific, and its main function is to relay (“route”) the received signal forward so that the nodes, which are not in the radio coverage of the original sender can also receive it. This kind of “radio relay routing” solution is dependent on the radio technology applied, which means that it needs to be realized in a specific way for each different radio technology.

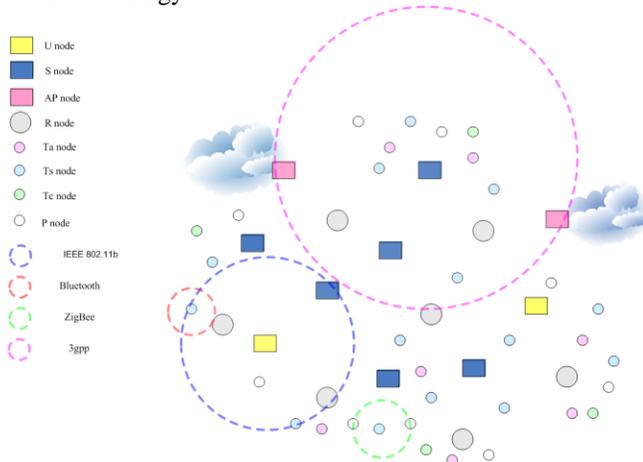


Figure 1. System model of a heterogeneous wireless network.

Some clusters of the network may need a specific method and optimized algorithm for physical level routing. Such optimization may be needed, for example, because of the limited power capabilities of the sensor nodes. For some network clusters such ad hoc routing protocol, like AODV, may be good enough; however, some of the nodes such as very limited capability sensor networks may require more optimized ad hoc routing protocol in the sense of memory and battery consumption. In addition, it may be more efficient to have a proactive protocol in operation when the network cluster is more static and not mobile. This means that the heterogeneous wireless network may consist of network clusters applying different physical routing algorithms. Therefore, several different physical ad hoc routing methods and protocols should be supported. When several different radio access and physical routing protocols are integrated into a single system, interoperability will be very big challenge. As a solution for interoperability, the overlay approach has been used in this work.

Thus, our hierarchical networking concept relies on the overlay approach, in which the radio relay routing, physical ad hoc routing and overlay routing are executed on top of each other, as in Figure 2. The overlay routing is applied on top of the physical networking and radio access specific networking. In the overlay routing, the application level messages are stored in packets called bundles, which are routed between logically neighboring overlay routers, e.g., a, b, and c. There can be several physical routers between the referred neighboring overlay routers, for example 1-5. And there can also be several radio relays between physical routers respectively. However, radio relays are beyond the scope of this paper.

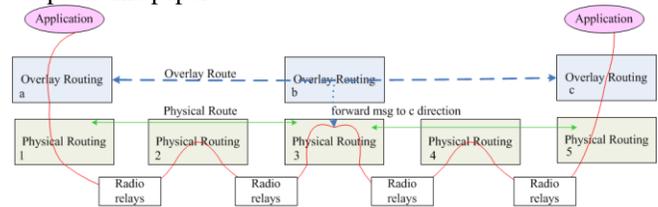


Figure 2. An example of hierarchical routing configuration.

IV. HIERARCHICAL ROUTING

Hierarchical routing is analyzed in this chapter with the aid of network graphs. A novel hierarchical routing algorithm is provided, based on the reasoning. Finally, a procedure for the hierarchical search and network optimization is discussed.

A. Network Graphs

An example of a heterogeneous wireless network system is shown in the form of a graph in Figure 3. In the example, a physical network graph (G_{PN}), vertex ($V_{PN}=0$) i.e., node (0) represents the User node. Each vertex has certain characteristics such as location (L), overlay routing capabilities (OR), physical routing capabilities (PR), radio capabilities (R), power capabilities (P) and computing power (Cp), $V_{PN}\{L, OR, PR, R, P, Cp\}$. The edges (E_{PN}) represent

the possible physical communication links between two or more nodes. Each edge has certain characteristics such as, for example, distance (D) and delay (Δt), $E_{PN}\{D, \Delta t\}$. In the example, the overlay network graph (G_{ON}) is established by the U, and S vertices (V_{ON}). The dotted lines represent the edges of the overlay network (E_{ON}). The overlay network graph is here said to be a virtual graph of the physical network graph ($G_{ON} \subset G_{PN}$). Respectively, we can define the radio network graph (G_{RN}), which shows the radio network below the physical network ($G_{ON} \subset G_{PN} \subset G_{RN}$). Therefore, the system model is here said to be hierarchical.

The G_{PN} can be represented in the form of a (search) tree (T_{PN}) from the perspective of the $V_{PN}=0$, i.e., user node 0 (A), shown in Figure 4. Such a tree does not have cycles, and the source of the search is represented as the root of the tree ($T_{PN} (V_{PN}=0)$). A search path is a route from the root of the tree to the leaf of the tree, representing the destination of the search. Such a search tree can be created for each node of the G_{PN} respectively.

Respectively, G_{ON} can be represented in the form of a tree (T_{ON}) shown in Figure 5. It is easy to see that the height of the overlay network tree is smaller than the height of the physical network tree. This means that the overlay network

path from source to the destination usually contains a smaller number of hops.

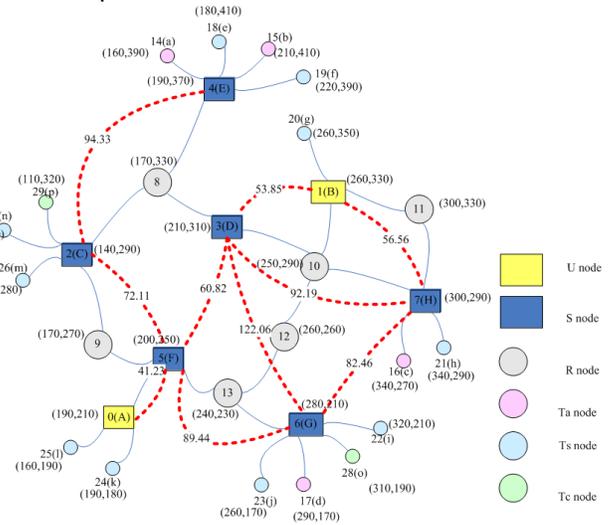


Figure 3. Example System Graph (G_{PN} and G_{ON}).

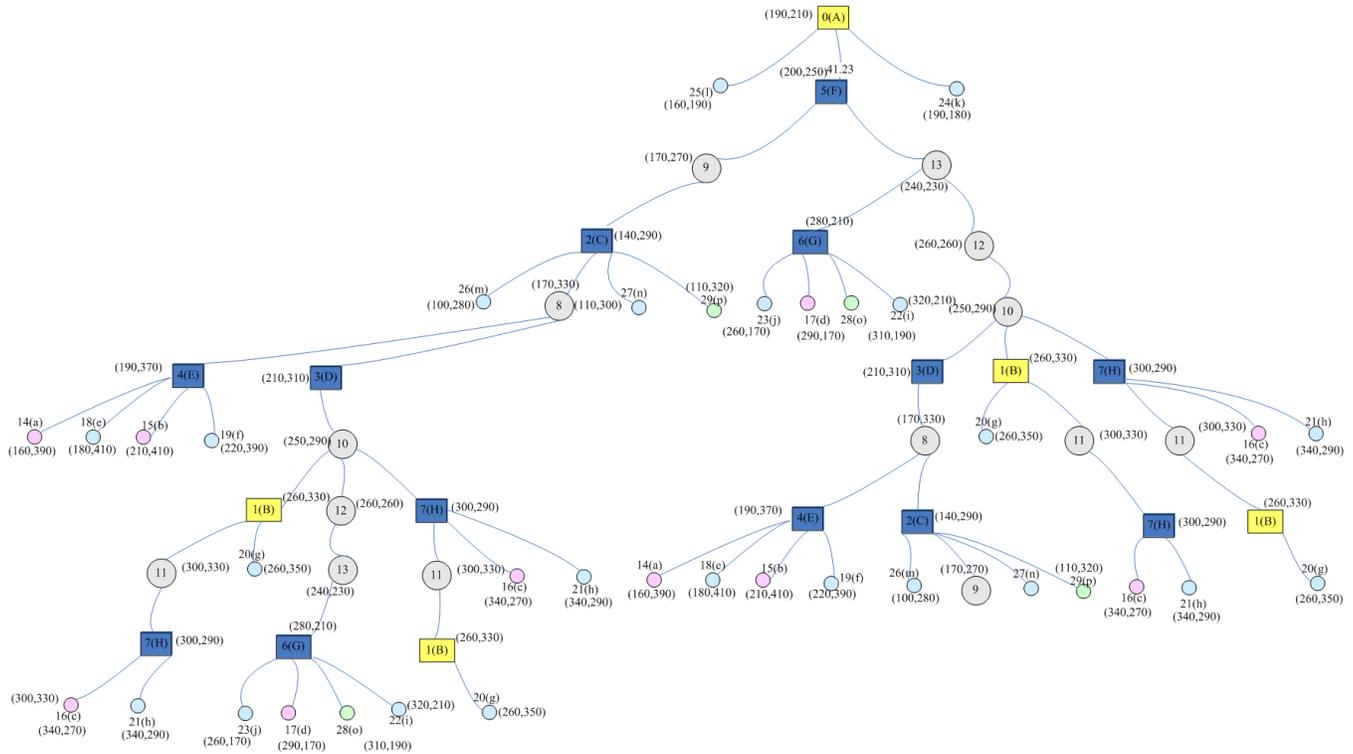


Figure 4. Example System physical network Tree (T_{PN}).

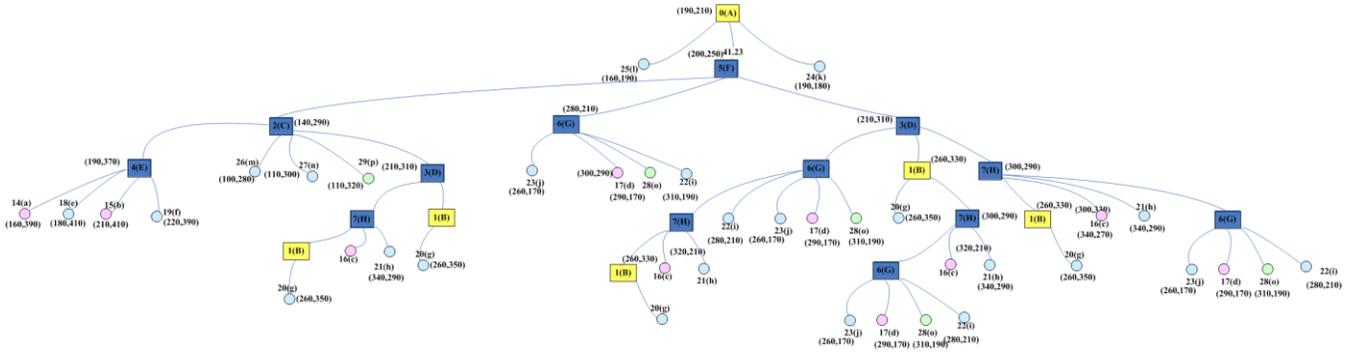


Figure 5. Example System Overlay Network Tree (T_{ON}).

B. Reasoning

The reasoning of the hierarchical search algorithm is represented in the following:

- Each edge in the search path means an additional communication delay for the search. Therefore, the number of levels in the search tree needs to be minimized. For example, if the search proceeds into deep sub trees, which do not have the destination, the search unnecessarily disturbs the vertices and consumes the radio bandwidth in the area of the leaf sub tree.
- Each vertex in the search tree processes the search, and it adds processing delay (Δt_p) to the search. Therefore, the number of vertices in the search path needs to be minimized. It can be claimed that the search unnecessarily disturbs all vertices in the search path, if the vertex is not the destination. Unnecessary disturbance of any vertex should be minimized.
- Let us call the minimization of the search tree levels, minimization of the number of vertices in the search path and minimization of vertex disturbance *search tree minimization*.
- The number of levels in the search tree is lower for the T_{ON} than for T_{PN} . Therefore, it is assumed that the search tree can be minimized by relying on hierarchical search, in which the search is executed in the overlay level (T_{ON}) and the physical level search is limited to the discovery of the physical paths between each pair of neighboring S nodes (T_{PN} is split into sub trees). This also means that the hierarchical search is executed in T_{ON} (Figure 5.) and in the split sub trees of T_{PN} visualized in Figure 6. In this way, the physical level search results in a local optimum physical path, called a *logical short-cut*, between neighboring S/U nodes, and the overlay level search results optimum path between source and destination (S/U or T_* nodes).
- Some of the vertices are more powerful than others, for example, some can have good power sources and a good computing platform while others may be battery-operated. It is clear that powerful vertices are better nodes for routing. Therefore, they are preferable nodes

- in the search path, and the usage of limited capability nodes (bottlenecks) will be minimized.
- When looking at different search paths in G_{PN} , T_{PN} it is assumed that removing the bottleneck nodes from the search path reduces the total communication delay (Δt_c) of the search. Let us here call the removal process *network optimization*.
- The network optimization process is focused on the split sub trees of T_{PN} ; see Figure 6. Because, the R nodes are assumed to be the bottleneck nodes, the S/U nodes actively try to remove them from the local physical communication paths, and create a *physical short cut* between the neighboring S/U nodes. As a result of successful network optimization, the search tree is like the T_{ON} visualized in Figure 5.

Summarizing, the hierarchical search with search tree minimization and network optimization processes results in a situation, where the search path consists only of powerful and well-connected S/U nodes and not bottleneck nodes.

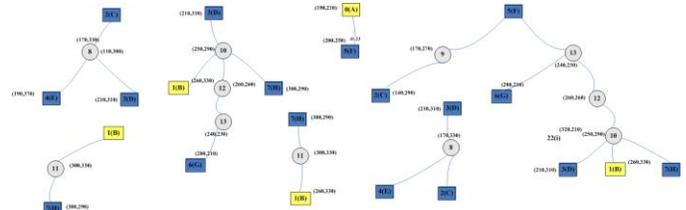


Figure 6. Split sub trees of T_{PN} .

C. Hierarchical Search Algorithm

The hierarchical search algorithm is represented in Figure 7. When the power of a U/S node is switched on, the device will broadcast *DiscoverReq* to all of its neighbors with the information of the node itself. Each overlay node receiving the *DiscoverReq* stores its key contents and replies with *DiscoverRep*, which is sent in a unicast manner to the source of the *DiscoverReq*. The *DiscoverRep* contains overlay level routing and service information, which will be delivered to the original source of the *DiscoverReq*. Sending the *DiscoverRep* triggers searching for the physical route

between the neighboring overlay nodes, for example using the AODV RREQ/RREP procedure. When the original source receives *DiscoverRep* via the discovered physical route, the system has established a *logical short cut* between the neighboring overlay nodes, and is ready to provide messaging services for applications.

When an application message (*APP-msg*) is received from the upper layer and the overlay route is known, it is forwarded towards its intended destination. Otherwise, an overlay route is searched first, and then the message is forwarded towards the destination. In this manner, the application message will be delivered to the destination using hierarchical search. At any time after the system is ready, the network optimization can be initiated. In the network optimization, direct wireless communication links for the neighboring overlay nodes may be created as *physical short-cuts* in the cases where it is physically possible with the available radio access technologies of the overlay nodes.

```

Algorithm HI-Search /* Hierarchical Search */

1. WHEN n(OFF) → n(ON) THEN
2.     send (DiscoveryReq, Bcast)
3. WAIT until receive (Msg)
4.     SWITCH Msg
4.         CASE DiscoverRep (ucast)
5.             store (DiscoverRep)
6.             start (timer, Net-Opt)
7.         CASE DiscoverReq (Bcast)
8.             IF n == ON THEN
9.                 store (DiscoverReq)
10.            send (DiscoverRep, Ucast)
11.            ELSE
12.                update (DiscoverReq)
13.            forward (DiscoverReq, Bcast)
14.         CASE applicationMsg
15.            IF no route THEN
16.                send (ON-RouteReq)
17.            IF route THEN send APP-msg
18.         CASE ON-RouteReq
19.            IF n == destination THEN
20.                send ON-RouteRep
21.            ELSE forward ON-RouteReq
22.         CASE Timeout (Net-Opt)
23.            optimize (network)
24.            start (timer, Net-Opt)
24.     ENDSWITCH
25. ENDWAIT
    
```

Figure 7. Hierarchical Search Algorithm.

D. Procedure of the Hierarchical Search

The basic procedure of the hierarchical search algorithm is shown in Figure 8. First, after power on, each overlay node initiates the logical neighbor discovery procedure by sending *DiscoveryReq* messages to indicate their presence to their

neighbors. Based on these broadcast messages, the physical routers in the chain can add the information about their physical neighbors into their routing tables. These messages are forwarded by all the nodes until an overlay node receives them. When an overlay node receives the *DiscoverReq*, unicast sending of *DiscoverRep* to the source of the *DiscoverReq* is activated. This activates searching of physical routes between the overlay node, and neighboring source overlay node.

The network may consist of different routing clusters, clouds in Figure 8., which may apply different physical routing protocols. For example, in cluster 1, the AODV route discovery will be executed, and as a result, a physical route between A and C can be discovered. The other clusters may use any other routing protocol for route discovery. When the physical route has been discovered, then the *DiscoverRep* is sent to the source overlay node. As a result of the logical neighbor discovery procedure, the overlay nodes know their physical and logical overlay neighbors and the *logical short-cut* has been established between the overlay neighbors. The physical routes between logical neighbors are stored in the physical level routing tables. After this phase, network optimization may be activated.

When an application message (*APP-msg*) is received from upper layers, it triggers searching of the overlay route by sending *ON-RouteReq* towards the logical neighbors. Each intermediate overlay node forwards the *ON-RouteReq* until the destination is discovered. The destination node then replies with *ON-RouteRep* message, which is sent via the same route, which the *ON-RouteReq* used. The nodes in the path update the routing tables accordingly to enable smooth forwarding of the *APP-msg* from source to destination i.e., from A to E.

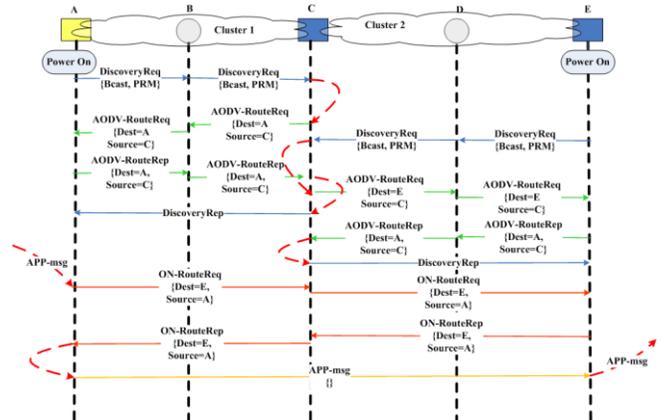


Figure 8. Hierarchical routing procedure.

E. Network Optimization

In the network optimization, direct wireless communication links between the neighboring overlay nodes are created (See Figure 7. row 22, and dotted red links in Figure 3.). These links are called *physical short-cuts*, and they can be created in the cases when the overlay nodes can apply larger transmit power or use a longer distance radio

access method to communication with the neighboring overlay node(s) directly.

Because of these physical short-cuts, the physical route can skip some of the physical routers, which makes the path shorter compared with a communication without them. For example, the physical communication path without the physical short cuts between B and D can consist of 3 intermediate physical hops via physical routers numbered 4, 6 and 7 (see Figure 3.). In that case, the logical short-cut between B and D is available via the referred intermediate hops, and it can be used to make long -distance routing more efficient. However, enabling also a physical short-cut link could enable node B to reach node D via a direct radio link without any intermediate physical routers by using, for example, a somewhat larger transmit power or other radio access system.

It is here assumed that the overlay node is a higher capability node, which usually has more power capabilities and can also have several different radio access technologies to be used for communication. Therefore, such nodes are able to create referred physical short-cuts. In addition, it is assumed that such nodes are able to act as cluster heads in the network topology. Therefore, the number of overlay nodes can be used as a measure of *clustering level* in the system. If there is smaller number of cluster heads, i.e. overlay nodes, then there are not many clusters in the network. If there are more cluster heads, then there are more clusters in the network.

Let us define low degree (D_L) to indicate the number of nodes, which have a small (0-2) number of neighbors. Usually, these kinds of nodes are other than overlay nodes, because those nodes usually have a limited number of radio accesses and power capabilities. Respectively, high degree (D_H) indicates the number of nodes, which have higher (> 2) number of neighbors. Usually, these kinds of nodes are overlay nodes i.e., cluster heads. The degree of clustering (D) is here defined as a function (1) depending on the number of low and high degree nodes, and it is used to indicate the level of clustering in a specific topology (T) in a specific moment of time (t).

$$\Delta(T, \tau) = \Delta_H(T, \tau) / \Delta_L(T, \tau). \quad (1)$$

The degree of clustering (1) is larger when the number of high degree nodes increases, and smaller when there are fewer high degree nodes. When the number of low degree nodes is significantly larger than the number of high degree nodes, the system represents a scale-free network. Then a majority of nodes have very few neighbors, and only a few nodes have many neighbors. Usually, the heterogeneous wireless network represents this kind of scale-free phenomenon. Because the degree of clustering depends on the topology and time, the effect of physical short-cuts for the path lengths and performance are in this work studied by means of simulations .

V. EVALUATION

Evaluation of the hierarchical search algorithm, network optimization and related procedures is provided in this chapter.

A. Evaluation of Hi-Search Algorithm

The depths of the search paths for the example graph shown in Figure 3. are shown in Figure 9. There are 37 possible search paths for both physical and overlay networks, see Figures 4 and 5 respectively. Each search path is shown in the x-axis, and the depth of the search path is shown in the y-axis in Figure 9. For example, for search path number 11, the depth of the physical search path is 10 and the depth of the overlay search path is 5. In general, the search path depths for the overlay routes are lower than the search path depths for the end-to-end physical routes. The *Hi-Search* algorithm provided applies overlay route search, which means lower search paths.

The physical search path depths of overlay hops are shown in Figure 10. (See also Figure 5.). The y-axis shows the physical search path depths, and the x-axis shows the number of their required searches in Figure 3. in a physical routing situation. For example, the physical search path 5-9-2, whose depth is 2, happens 17 times in a physical routing situation. The referred physical search paths seem generally to happen multiple times in the example network in a physical routing situation. This is not very efficient, and therefore the algorithm creates logical short-cuts between the neighboring overlay nodes. Then there is a need to execute referred physical search paths only once for the network, and network optimization can be based on it. The referred network optimization action is initiated in row 22 of the *Hi-Search* algorithm to check and create possible physical short-cut link between the neighboring overlay nodes.

The number of control message sending actions is shown in the y-axis of Figure 11. When the physical route between neighboring overlay nodes is searched initially and optimized, the number of control message send actions is at about the same level as in physical routing ($x=7$). However, after the optimization has been executed, then the number of control message send actions drops significantly, because there is no need to repeat optimization. It can be seen that the number of control message send actions is lower when applying the *Hi-Search* algorithm compared with physical routing.

The total delay in the search is shown in the y-axis of Figure 12. It is assumed here that the delay in each physical hop, i.e. the radio link, is 10ms, the optimization happens in a parallel manner and the processing delay in each node is zero. The peaks of the delay for the *Hi-Search* algorithm are related to optimization of the network. After optimization, the delays are at a lower level. As a result, it is seen that the *Hi-Search* algorithm is better because it has lower search delays than physical routing.

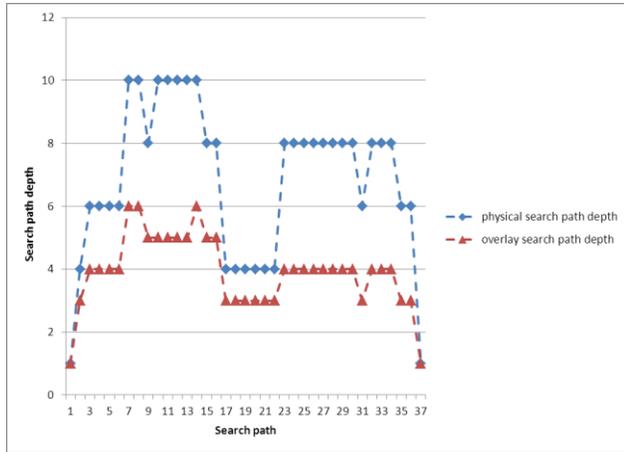


Figure 9. Search path depths.

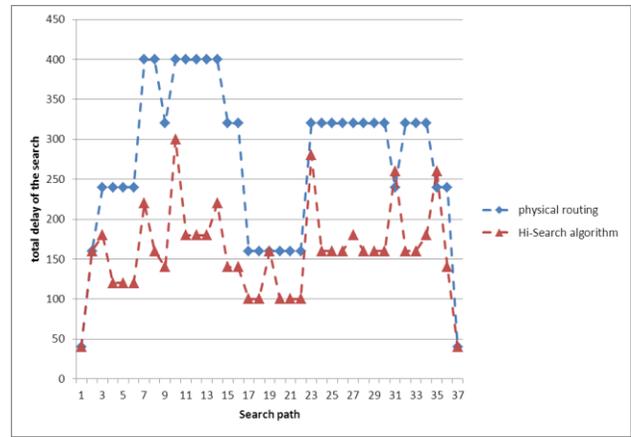


Figure 12. Delay the search.

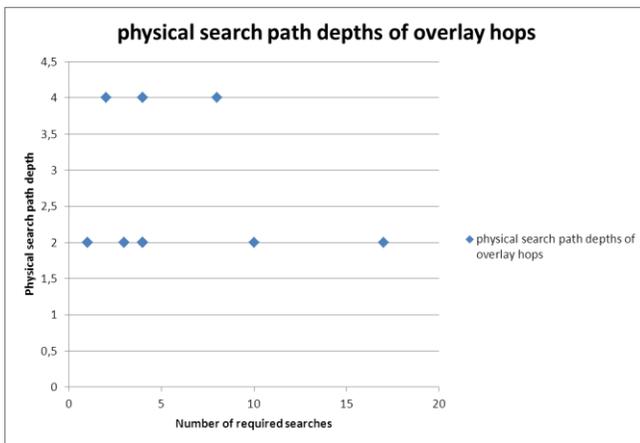


Figure 10. Physical search path depths of overlay hops.

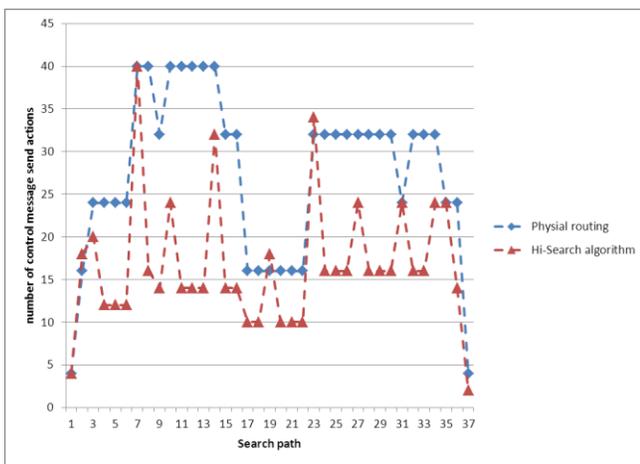


Figure 11. The number of control message sending.

In practical situations, the physical characteristics including the delay in each edge vary according to applied radio access technology. The network optimization removes weak and high delay edges from the path, which may make the delay difference between physical search and *Hi-Search* algorithm even larger than what is shown in Figure 12. In addition, the processing delay of each vertex is usually bigger than zero. When applying the *Hi-Search* algorithm, the number of intermediate hops in the path is minimized in such a manner that weak nodes are removed from the path by network optimization. Therefore, in a practical situation the delay difference between physical search and *Hi-Search* algorithm is even larger than what is shown in Figure 12.

B. Evaluation of network optimization

The evaluation of network optimization has been carried out with NS-2 simulations to evaluate its effects to the end-to-end delays, physical route lengths and throughput. In addition, the effects of degree of clustering, i.e. the number of physical short-cuts to these, have been studied. Mobility is not allowed in the simulations, and the comparison is carried out in such a manner that the only changing factors are the number of physical short-cuts and the transmission power. In this way, it is expected that the effect is seen in pure manner.

Four different topologies have been simulated, each of which have a different number of nodes: 61, 100, 150 and 200. The applied simulation parameters are shown in Figure 13. The physical level routing solution is called eAODV, and the overlay level solution is called eORCP.

Delay in sending a packet between source and destination as a function of the number of nodes is shown in Figure 14. The blue line represents eAODV routing in the network, in which all the nodes have transmission power P_t 0.002818, which means 2.818 mW and ca. 50m transmission range. In this case, there are no overlay nodes, which means that all the nodes are in the same cluster. The other lines represent eORCP with a different number of overlay nodes (2, 4, and 6) added into the same network topology. The overlay nodes have transmission power P_t 0.2818, which means 281.8 mW and approx. 150m transmission range. Therefore, the

overlay nodes can be connected with neighbor nodes in a larger neighborhood area. In the simulations, the end-to-end delay is an average of 50 measured round trip end-to-end delays. According to the simulation results, the end-to-end delay is shorter when the number of overlay nodes increases. The differences in the delays of eAODV and eORCP-* cases are not very big, however; the simulations give a clear indication that the larger number of physical short-cuts makes the end-to-end delay shorter.

```

set val(chan) Channel/WirelessChannel ;# channel type
set val(prop) Propagation/TwoRayGround ;#radio-propagation model
set val(netif) Phy/WirelessPhy ; # wireless
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# queue type
set val(ll) LL ;# Link layer type
set val(ant) Antenna/OmniAntenna ;# antenna type
....
# SharedMedia interface with parameters to make
# it works like the 914MHz Lucent WaveLAN DSSS radio interface

Phy/WirelessPhy set CPTthresh_ 10.0
Phy/WirelessPhy set CSTthresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6

#the range is about 50 meters

Phy/WirelessPhy set Pt_ 0.002818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0
    
```

Figure 13. Simulation parameters.

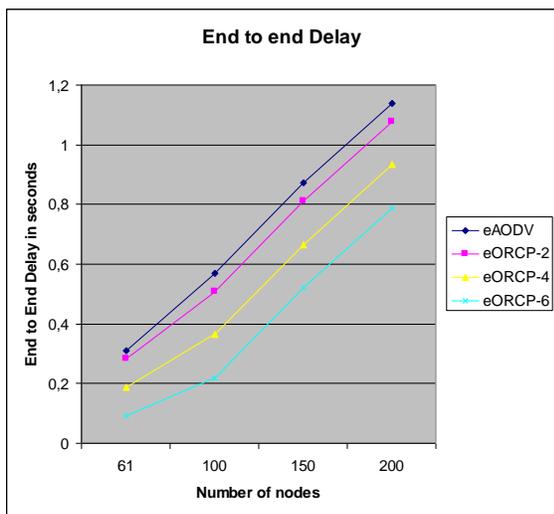


Figure 14. End-to-End Delay.

The reason for the shorter end-to-end-delay can be seen from Figure 15. Because the overlay nodes use larger transmits power when implementing the wireless short-cuts, they enable a shorter physical route to the destination. Because there is some delay in each of the wireless links, the

end-to-end delay is shorter when the number of hops is fewer. For example, in 61 node network, the end-to-end route for the pure physical router network (eAODV) consist of 55 hops, and when applying 6 overlay nodes, the physical route consist of 15 hops. This gives a clear indication that the larger number of physical short-cuts reduces the number of intermediate hops, i.e. the path of a route is shorter. However, it is obvious that the absolute quantity of reduction in the delay and the number of intermediate hops in the route depends on the topology.

Throughput in delivering a large number of packets between source and destination as a function of number of nodes is shown in Figure 16. In the measurement, the applied packet size has been 512 Bytes. As can be seen, the system with eAODV solution has a somewhat lower throughput compared with, for example, eORCP-2, eORCP-4 and eORCP-6. This means that the performance improves when the number of overlay nodes increases independent of the number of nodes attached into the system. Thus, the simulations give a clear indication that increasing the number of physical short cuts in the system improves system performance. The improvement is not very big, however. It is seen that this improvement is generic, even if it is obvious that the absolute quantity of the performance improvement depends on the topology.

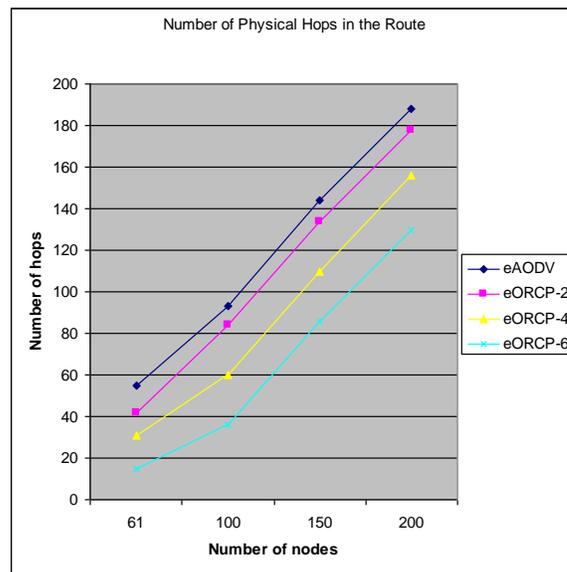


Figure 15. The number of physical hops in the route.

The degree is here used to indicate the level of clustering in a specific topology and moment of time. In addition, the number of overlay nodes is used to indicate the level of clustering, as described earlier. In our simulation cases, the overlay nodes have larger transmission power (P_t 0.2818, ~ 150m range); and it enable them to have more than 2 neighbors in the communication range in the simulated topologies. Instead, the physical router nodes have lower power (P_t 0.002818, ~50m range), and therefore they can

have only 0-2 neighbors. In the simulation cases, the number of overlay nodes (D_H : 0, 2, 4, 6) has been significantly smaller than the number of physical router nodes (D_L : 61, 100, 150 and 200). Therefore, the simulated topologies represent scale-free networks, because the majority of nodes have very few neighbors (D_L is big), and only a few nodes have many neighbors (D_H is small).

The discovered physical route lengths are shown in Figure 17. The simulation of 4 topologies all indicate that when the degree of clustering increases, the number of hops in the discovered physical route decreases. This result indicates typical small world phenomena, where the high clustering means shorter physical routes between nodes.

Throughputs of 4 topologies as functions of degree of clustering are shown in Figure 18. As can be seen, throughput increases when clustering increases. This means that the degree of clustering has a positive effect on the throughput. The system can be claimed to scale better because, when the clustering is higher, throughput is better and delay lower.

The simulation results indicate clearly that when increasing the number of physical short-cuts in the system, the end to end delays and the physical routes become shorter, and throughput improves. Thus, when the degree of clustering increases, the physical routes become shorter and

the performance of the system improves. Simultaneously, the system scalability is improved, because when the clustering is higher, throughput is better and delay lower. These improvements seem to be generic; however, it is obvious that the absolute quantity of the improvement depends on the topology.

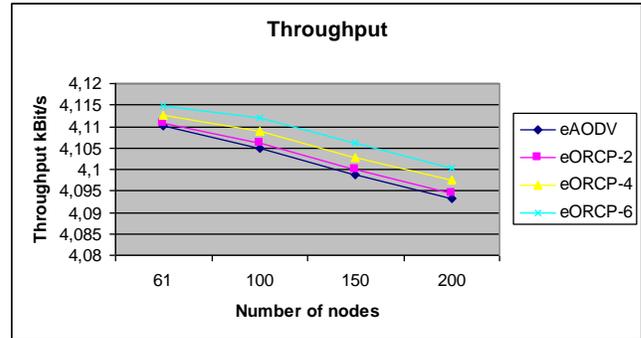


Figure 16. Throughput.

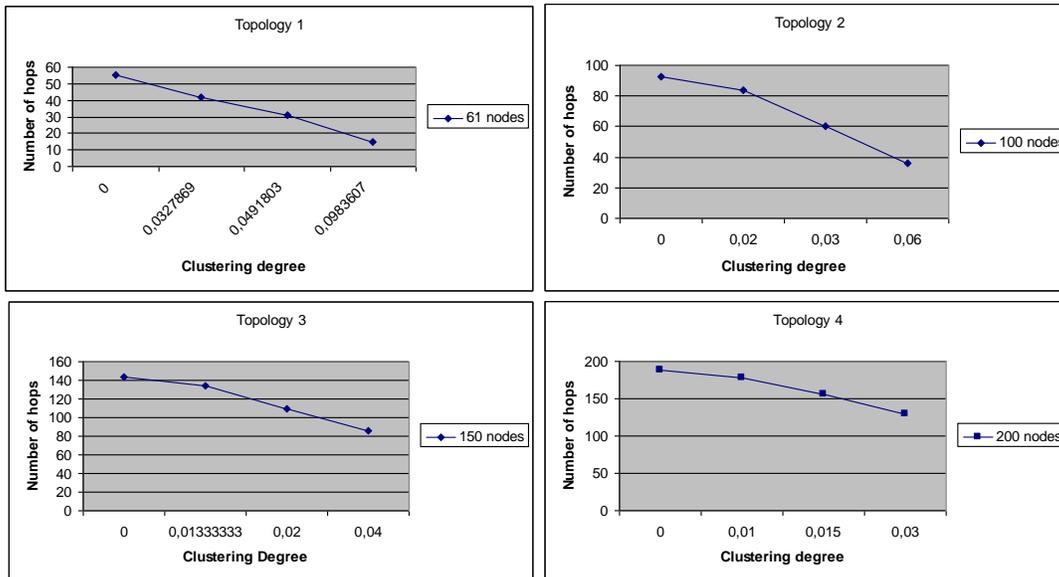


Figure 17. Route length as a function of degree of clustering.

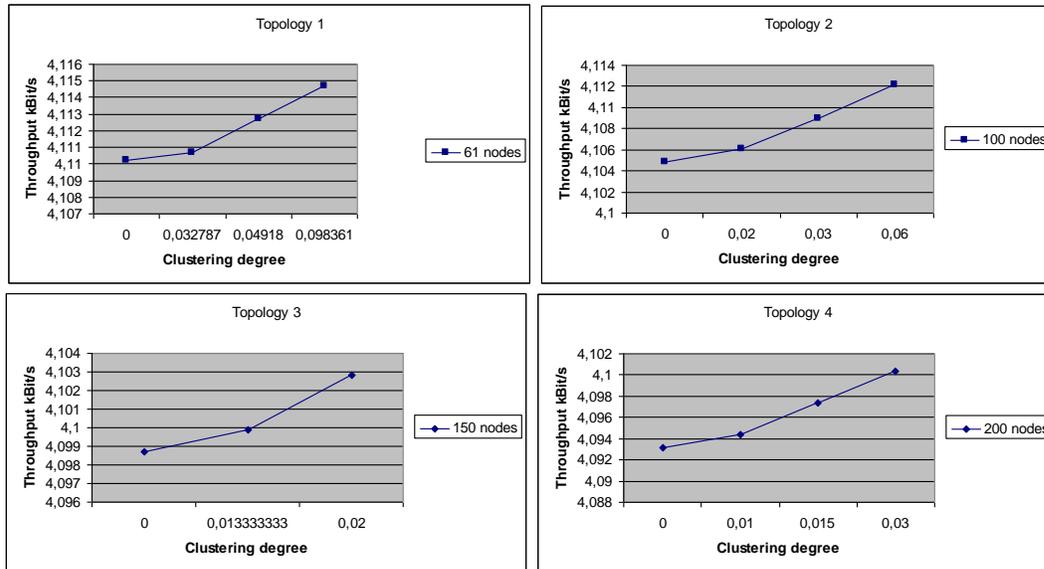


Figure 18. Throughput as a function of degree of clustering.

C. Evaluation of Procedures

The topology of the simulated network is shown in Figure 3. The simulated network consist of two U nodes, six S nodes, six R nodes, ten sensor nodes (Ts), four actuator nodes (Ta) and two camera nodes (Tc). The blue lines represents physical routes, and the dotted lines shows overlay neighbor connections. The numbers represent the physical addresses of nodes, and letters of the alphabet represent overlay addresses. In this example, all the nodes have both physical address (1...30) and the service nodes (U, S, Ts, Ta and Tc) have a logical address (A ... H, a ... p). The numbers in parenthesis indicate the location of the nodes. The simulation parameters of the radio links are shown in Figure 19.

The procedure for simulation is described briefly in the following:

- **Initialization and startup of the nodes and their services:** During this process, all the layers of the nodes are started including the services of the nodes. The services include virtualized M2M services such as overlay router (P2P router), switch, heating regulator, temperature sensor, and surveillance camera.
- **Hierarchical Neighbor Discovery:** During this procedure, the physical router inside the nodes detects physical neighbors, and the logical overlay router inside some nodes becomes aware of its logical neighbors. The length of the logical short-cut, i.e. the intermediate wireless hops between the path of neighboring overlay nodes, and the delay between logical neighbors have a significant contribution to the efficiency of the neighbor discovery process.
- **Network optimization:** During this procedure, the network creates the physical shortcut. In the simulations,

the capabilities of the creation of physical short-cut are analyzed and evaluated in a functional sense.

- **Service Discovery:** During this procedure, the user is searching via the U node for all the services, which are available to him/her at the time of the search. The list of all the available services is shown as a result of the search. The number of discovered services and the waiting time of the search have an essential meaning for the user.
- **Service use:** During service use, service level payloads are transferred from the service node to the user node. Measuring the end-to-end delay, the number of physical intermediate hops in the route and throughput is used in the evaluations.

The measured delays in hierarchical neighbor discovery are shown in Figure 20. The delay values are shown on the Y axis in seconds as a function of intermediate physical hop numbers. In the simulated topology, there were only 1, 2 or 4 physical hop routes between the overlay neighbors. The delays represent time from the sending of NeighborHelloReq (DiscoveryReq in step 2 of Figure 7.) to receiving NeighborHelloRsp (DiscoverRep in step 4 of Figure 7.), i.e. the creation of logical short-cuts. The delay includes discovery of the physical route to the logical neighbor, and delivery of the related messages using the route. The measurements indicate that the number of physical hops increases the average delay in the hierarchical neighbor discovery. However, the variance in the measured delays in the hierarchical neighbor discovery is quite a high. The reason for this is assumed to be the loss of messages in the simulated radio channel (Propagation/TwoRayGround) or message drops in the physical router queue (Queue/DropTail/PriQueue).

```

set val(chan) Channel/WirelessChannel      ;# channel type
set val(prop) Propagation/TwoRayGround    ;# radio propagation model
set val(netif) Phy/WirelessPhy           ;# wireless
set val(mac) Mac/802_11                  ;# MAC type
set val(ifq) Queue/DropTail/PriQueue     ;# queue type
set val(ll) LL                            ;# link layer type
set val(ant) Antenna/OmniAntenna        ;# antenna type
...
# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1 meter above it
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 0.95
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPTthresh_ 10.0
Phy/WirelessPhy set CSTthresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6

# Transmitter power is divided by 100 for the smaller nodes.
# The range is about 50 meters
Phy/WirelessPhy set Pt_ 0.002818
Phy/WirelessPhy set freq_ 914e+6
set val(chan) Channel/WirelessChannel      ;# channel type
set val(prop) Propagation/TwoRayGround    ;# radio propagation model
set val(netif) Phy/WirelessPhy           ;# wireless
set val(mac) Mac/802_11                  ;# MAC type
set val(ifq) Queue/DropTail/PriQueue     ;# queue type
set val(ll) LL                            ;# link layer type
set val(ant) Antenna/OmniAntenna        ;# antenna type
...
# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1 meter above it

Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 0.95
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface

Phy/WirelessPhy set CPTthresh_ 10.0
Phy/WirelessPhy set CSTthresh_ 1.559e-11
Phy/WirelessPhy set RXThresh_ 3.652e-10
Phy/WirelessPhy set Rb_ 2*1e6

# Transmitter power is divided by 100 for the smaller nodes.
# The range is about 50 meters

Phy/WirelessPhy set Pt_ 0.002818
Phy/WirelessPhy set freq_ 914e+6
Phy/WirelessPhy set L_ 1.0
    
```

Figure 19. The simulation parameters.

The number of discovered services and the waiting time of the search have an essential meaning for the user. The discovered services are shown as printed output from the U nodes 0 and 1 in a simulation execution in Figure 21. In the example simulation execution, 6 services were discovered out of 16 possible. The service reply messages from the P2P router C and D and sensor nodes j, h and g were dropped in the simulated radio channel, and therefore the services

behind them were not discovered. The measured waiting time of the service discovery results was on average 0.4573 sec. The example service discovery simulation case indicates that loss of messages in the wireless channels causes undiscovered services unless reliable delivery services are not provided by the communication layer for the services layer.

During service use, service level payloads are transferred from the service node to the user node. The measured performance of simulated service use is shown in Tab I. When increasing the sending power, the number of intermediate hops decreases. For example, in our topology visualized in Fig. 6, the number of intermediate nodes between U-nodes 0 and 1 was reduced from 5 to 3. As a result of this, the end-to-end delay was decreased from 30.2 ms to 17.9 ms. In addition; throughput is also improved somewhat, from 4.1268 kbit/s to 4.1272 kbit/s. The measured performance of simulated service use indicates that the establishment of wireless short-cuts can be very useful, because it reduces the number of intermediate hops, makes end-to-end delay shorter and improves throughput.

Simulation of *dynamic* network optimization proved to be very challenging with the NS-2 simulator, because it did not seem to be possible to change the transmission power or applied radio technology dynamically after the node had been created in the simulator.

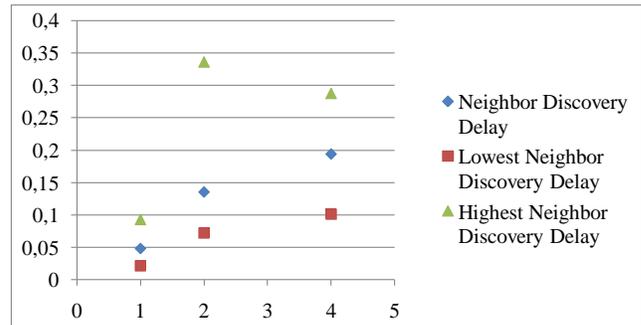


Figure 20. Average delays in seconds in hierarchical neighbor discovery.

```

***** Node 0's Services at 200.00 *****
-- Location / Updated / Service --
-- l / 98.22 / H_sensor_A2 --
-- k / 96.21 / T_sensor_A1 --
-- F / 150.12 / P2P_Router4 --
-- G / 150.40 / P2P_Router5 --
-- d / 150.40 / H_Regul_G1 --
-- i / 150.40 / T_sensor_G2 --
-- o / 150.40 / S_camera_G4 --
-- H / 150.44 / P2P_Router6 --
-- c / 150.44 / H_Regul_H1 --
-- D / 150.45 / P2P_Router2 --
*****
***** Node 1's Services at 200.00 *****
-- Location / Updated / Service --
-- g / 88.17 / A_sensor_B1 --
*****
    
```

Figure 21. The discovered services in the U – nodes.

TABLE I. MEASURED PERFORMANCE OF SIMULATED SERVICE USE

Sending power Pt	Number of intermediate hops	End to end delay ms	Throughput kbit/s
0.009818	3	17.9	4.1272
0.002818	5	30.2	4.1268

D. Discussion

The problem in flat route discovery is that search queries are also forwarded in the deep leaves of the search trees. This problem is solved in the hierarchical routing in the sense that only the nearest logical overlay nodes are initially searched at the physical route level. The result of this step is discovered physical routes between neighboring overlay nodes. After this phase, the network can be optimized by removing non-optimal radio links and physical routers from the referred local physical path. The result of this step can be direct connection between neighboring overlay nodes, which may be most optimal for local communication. When an application message needs to be sent, then searching of the end-to-end route is triggered. If network optimization has been successful, then the search paths depths are as in overlay search, i.e. significantly lower than the search path depths for the end-to-end physical routes. The evaluations also indicate that then the number of control message send actions and delay of the search are also lower. In addition, the search queries do not unnecessarily disturb the nodes, which are in the deep leaves of the search trees.

The measurements of hierarchical neighbor discovery simulations indicate that the number of physical hops increases the average delay in the hierarchical neighbor discovery, but the variance is quite high because of message losses in the communication channel. The loss of messages also causes undiscovered services when no reliable communication is provided by communication layer to the services layer. The measured performance of simulated service use indicates that the establishment of wireless short-cuts can be useful, because it decreases the number of intermediate hops, makes end-to-end delay shorter and improves throughput.

The evaluations of the hierarchical routing have been carried out in multiple steps: theoretical evaluation of the Hi-Search algorithm, simulation of the network optimization and simulation of the procedures. The theoretical evaluation is limited in the sense that only one example network has been represented; however, the aim is to enlarge and generalize the graph theoretical evaluation in the next step. Limitations of the NS-2 environment cause serious challenges in simulation-based evaluation of network optimization and procedures. This is because it is not possible to simulate properly the features of dynamic wireless networks, such as, for example, changing the transmission power, changing applied radio technology dynamically after the node has been created, and having more than one different radio and network interfaces for a single node. Therefore, evaluation of the network optimization and procedures was limited here to quite simple

topologies without any mobility. The aim in the next step is to simulate more complicated dynamic networks, more complex topologies, mobility and advanced features of hierarchical network with NS-3, and also to evaluate in a real experimental case.

VI. CONCLUSIONS

The evaluation indicates that the search path depths for the *Hi-Search* algorithm are lower than the search path depths for the end-to-end physical routes. The logical short-cuts, i.e. the physical routes between logically neighboring vertices, are searched only once, which reduces the number of required control message send actions. The search delays are lower compared with physical routing. The network optimization removes weak and high delay edges and vertices from the path, which may make the delay difference between physical search and *Hi-Search* algorithm even greater. The evaluation of network optimization indicates that increasing the number of referred physical short-cuts reduces the end-to-end delays, makes the physical routes shorter, and also improves throughput. When the degree of clustering increases, the physical routes become shorter and the performance of the system improves. The detected evaluation results of the network optimization with physical short-cuts conforms quite well to the phenomenon of small world and scale-free networks. The evaluation of procedures indicates that the average delays in neighbor discovery are increased by the number of physical hops. In addition, message losses in the radio channel increases variance in the neighbor discovery delays. Generally speaking, the service discovery delays were at a feasible level in the simulated topology. However, loss of messages in the wireless channels causes undiscovered services. The measured performance of simulated service use indicates that the establishment of physical short-cuts can be useful, because it reduces the number of intermediate hops, makes end to end delay shorter and improves throughput.

Summarizing, the evaluation results indicate that the *Hi-Search* algorithm with network optimization is able to lower search delays, make the physical routes shorter, and also improve throughput. In addition, solving the complexity and heterogeneity problems is made possible by localizing route search and abstracting communication to two different routing layers. However, because of practical limitations with the applied simulation platform, it was not possible to simulate properly dynamic features of different topologies and mobility. Therefore, the aim in the next step is to work with more complicated dynamic networks, more complex topologies, mobility and advanced features of the hierarchical network with NS-3, and also to evaluate hierarchical routing in a real experimental platform.

ACKNOWLEDGMENT

We would like to thank Tekes and VTT for funding this work.

REFERENCES

- [1] Latvakoski J., A Hierarchical routing algorithm for small world wireless networks. The Fifth International Conference on Communication Theory, Reliability, and Quality of Service, CTRQ 2012. 6p.
- [2] Latvakoski, J. Towards hierarchical routing in small world wireless networks. The Fifth International Conference on Wireless and Mobile Communications ICWMC 2009.
- [3] Latvakoski, J. Hierarchical routing concept for small world wireless networks. 6 p. ICWMC 2010: The Sixth International Conference on Wireless and Mobile Communications September 20-25, 2010 - Valencia, Spain.
- [4] Latvakoski, J. and Aapaoja, T. Towards a Routing Overlay for a Mobile Ad hoc Network. 6p. First International Workshop on Convergence of Heterogeneous Wireless Networks (ConWiN) 10th Jul 2005 Budapest, Hungary. 8p.
- [5] Latvakoski J., Aapaoja T., and Kärnä J. Evaluation of routing overlay solution for a Hybrid Mobile ad hoc networks. 12p. ERCIM Emobilty workshop. 28-30 May 2008 Tampere, Finland. 12p.
- [6] Milgram S. The small world problem. *Psychol. Today* 2, Pp 60-67. 1967
- [7] Watts D. and Strogatz S. Collective Dynamics of small world networks. *Nature* Vol 393. Pp 440-442. 1998.
- [8] Korzun, D. and Gurtov, A. 2011, "Survey on hierarchical routing schemes in "flat" distributed hash tables", *Peer-to-Peer Networking and Applications*, vol. 4, no. 4, pp. 346-375.
- [9] L. Adamic, "The small world web," in *Proc. Eur. Conf. on Digital Libraries(ECDL)*, Sept. 1999, pp. 443-452.
- [10] A. Broder, R.Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata A. Tomkins, and J. Wiener, "Graph structures in the web," *Computer Networks*, vol. 33, pp. 309-320, June 2000.
- [11] R. Albert, H. Jeong, and A. Barabasi, "Diameter of the world wide web," *Nature*, vol. 401, pp. 130-131, 1999.
- [12] Bettstetter C, (ed) *Self-Organization in Communication Networks: Overview and State of the Art*. Wireless world research forum white paper. Version 1.2. Aug 11, 2005. 44p.
- [13] Helmy, A. Small Worlds in Wireless Networks. *IEEE Communications Letters*, Vol 7. No 10. October 2003.
- [14] Verma, C.K., Tamma, B.R., Manoj, B.S., and Rao, R. 2011, "A Realistic Small-World Model for Wireless Mesh Networks", *IEEE Communications Letters*, vol. 15, no. 4, pp. 455-457.
- [15] Orier R., Templin F., and Lewis M. 2004. Topology Dissemination based on reverse-Path Forwarding (TBRF). IETF RFC 3684. Feb 2004.
- [16] Perkins C., Royer-Belding E., and Das S. RFC 3561. Ad hoc On-Demand Distance Vector Routing. IETF. Jul 2003. <http://tools.ietf.org/html/rfc3561>, Available 26th Nov 2012.
- [17] Clause T. and Jacquet, P. (eds). Optimized Link State Routing Protocol (OLSR). IETF RFC 3626. <http://www.ietf.org/rfc/rfc3626.txt>. Available 26th Nov 2012.
- [18] Ian D. Chakeres and Charles E. Perkins Dynamic MANET On-Demand Routing Protocol. IETF Internet Draft, draft-ietf-manet-dymo-12.txt, February 2008 (Work in Progress). Retrieved 2, 2012.
- [19] Luciana Pelusi, Andrea Passarella, and Marco Conti. (2006). "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad hoc Networks." *IEEE Communications Magazine* Nov 2006: Pp. 134-141.
- [20] V.Cerf (ed) (2007) Delay-Tolerant Networking Architecture. IETF RFC 4838. Apr 2007.
- [21] Latvakoski, J. and Hautakoski, T. Situated Message Delivery for Opportunistic Networks. 9p. ICT Mobile and Wireless Communications Summit 2008. 10-12 Jun 2008. Stockholm/Sweden.
- [22] Sylvia Ratnasamy, P. F., Mark Handley, and Richard Karp (2001). "A Scalable Content-Addressable Network." SIGCOMM'01, Aug 27-31, San Diego, USA: Pp. 161-171.
- [23] Ion Stoica, R. M., David Krager, M. Frans Kaashoek, and Hari Balakrishnan (2001). "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications." SIGCOMM'01, Aug 27-31, San Diego, USA
- [24] Ben Y. Zhao, L. H., Jeremy Stribling, Sean C.Rhea, Anthony D. Joseph, and Jon D. Kubiawicz (2004). "Tapestry: A Resilient Global-Scale Overlay for Service Deployment." *IEEE Journal on Selected Areas in Communications* Vol 22(No 1, January 2004): Pp. 41-53.
- [25] Antony Rowstron, P. D. (2001). "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems." *Proceedings of 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*. Heidelberg,D.
- [26] Nicholas J.A.Harvey, M. B. J., Stefan Saroiu, Marvin Theimer, and Alec Wolman (2003). "SkipNet: A Scalable Overlay Network with Practical Locality Properties." *Proceedings of USITS Seattle, WA. Mar 2003*: 14.
- [27] Ben Y. Zhao, Y. D., Ling Huang, Anthony D. Joseph, and Jon D. Kubiawicz (2002). "Brocade: Landmark Routing on Overlay Networks." *Proceedings of 1st International Workshop on Peer-to-peer Systems, IPTPS'02*: 6p.
- [28] Arturo Crespo and H. G.-M. (2002). "Semantic Overlay Networks for P2P Systems." *Computer Science Department, Stanford University. CA USA*: 15p.
- [29] Liu, X., Guan, J., Bai, G., and Lu H. 2009, "SWER: small world-based efficient routing for wireless sensor networks with mobile sinks", *FRONTIERS OF COMPUTER SCIENCE IN CHINA*, vol. 3, no. 3, pp. 427-434.
- [30] Belding-Royer, E.M. 2002, "Hierarchical routing in ad hoc mobile networks", *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 515-532.
- [31] Helmy A., Garg S., and Nahata N. CARD: A contact-based Architecture for Resource Discovery in Wireless Ad hoc Networks. *Mobile networks and applications* 10, pp. 99-113. 2005. Springer-Verlag.
- [32] Jiang C-J, Chen C., Chang J-W., Jan R-H., and Chiang T. C.. Construct Small Worlds in Wireless Networks using Data Mules. Pp 28-35. *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. Pp. 28-35.
- [33] Ken Y.K. Hui, John C.S. Lui, and David K.Y. Yau. Small world overlay P2P Networks. *Quality of Service, 2004. IWQOS 2004. Twelfth IEEE International Workshop on 7-9 June 2004*. Pp. 201 - 210.
- [34] Narten T., Nordmark E., and Simpson W. Neighbor Discovery for IP version 6. IETF RFC 2461. <http://www.ietf.org/rfc/rfc2461.txt>. Available 26th Nov 2012.
- [35] Clausen T., Dearlove C., and Dean J. MANET Neighborhood Discovery Protocol (NHDP), <http://tools.ietf.org/html/draft-ietf-manet-nhdp-04>, Expires December 31, 2007. Available 26th Nov 2012.

Design and Implementation of an Online XML Compressor for Large XML Files

Tomasz Müldner
Jodrey School of Computer Science
Acadia University
Wolfville, B4P 2A9 NS, Canada
e-mail: Tomasz.muldner@acadiau.ca

Jan Krzysztof Miziołek
IBI AL
University of Warsaw
Warsaw, Poland
e-mail: jkm@ibi.uw.edu.pl

Tyler Corbin
Jodrey School of Computer Science
Acadia University
Wolfville, B4P 2A9 NS, Canada
e-mail: 094658c@acadiau.ca

Christopher Fry
Jodrey School of Computer Science
Acadia University
Wolfville, B4P 2A9 NS, Canada
e-mail: chrisfry99@gmail.com

Abstract—Network-based applications using XML experience a performance penalty resulting from the verbose nature of this data format. This paper presents a novel XML-conscious compressor designed to alleviate these problems, using it for online compression and decompression. Two versions of the compressor were designed and implemented to find the most optimal solution and they were compared with offline compression/decompression. The tests show that for existing files online compression is less efficient than offline compression, however, online compression is superior for streaming or when compared to offline compression combined with sending the file through the network and subsequent decompression.

Keywords—XML; compression; network performance.

I. INTRODUCTION

The eXtensible Markup Language (XML) [16] is the most popular meta-language for the interchange and access of data. In particular, XML has been adopted as one of the main formats for online communications and Web applications. However, XML's markup and resulting verbose nature may increase the size of a dataset as much as ten-fold. For XML-based network applications, network bandwidth tends to become the bottleneck in the interchange of information; therefore these applications will experience a performance benefit from compressing XML data.

There has been considerable research on XML-conscious compressors, which unlike general data compressors can take advantage of the XML structure; see [2][3][4]. Most recently, there has been research on queryable XML compressors for which queries can be answered using lazy decompression, i.e., decompressing as little as possible when executing a specific query; see [5][6]. Also, there has been research on updateable XML compressors, for which updates can be saved without full decompression; see [7][8]. Online XML compressors are typically defined as compressors, which decompress chunks of compressed data whenever possible rather than processing it offline when the entire compressed file is available; see [9][10]. Clearly, for a compressor to be online implies that *only one pass* through

the document is required to compress it. This class of compressors is particularly useful for networked applications, specifically on networks with limited bandwidth. Numerous applications of XML use *streams*, abstract representations of sources/sinks, where the sources of data are dynamic and their contents are not known beforehand, e.g., measurements or logging. The contents are processed at run-time, either by XML Streaming Parsers, such as SAX [17] or StAX [24] or by ordinary text compressors such as GZIP [18]. Another approach is taken by Efficient XML Interchange (EXI) format [25], a compact representation of XML designed to reduce bandwidth requirements while maintaining efficient use of various resources such as memory and processing power (implemented using EXIficient [27]). While Snyder [26] determined that using EXI can double a bandwidth potential, it should be noted that EXI is not a queryable compressor.

This paper presents an online compression algorithm based on XSAQCT, an XML compressor developed by our group, see [11]. There are other online compressors, e.g., TREECHOP [12], but XSAQCT has a number of distinctive features, in particular it is queryable using lazy decompression, updateable [7], supports the streaming of data in a more compact representation than ordinary text compressors, and finally the structure of the XSAQCT's compression scheme allows a large reduction in processing time through parallelization on multi-core machines [13]. Possible educational applications of XSAQCT are described in [14]. Similar to TREECHOP, XSAQCT supports compression where the decompressor's output is the same as the original input (i.e., the document is semantically equivalent to the original document) or the output generates a *canonicalized* [15] XML document. Design of an early version of the compressor described in [1] did not support XML documents with mixed contents, attributes, or cycles, e.g., nodes with the consecutive children b, c and b. For example, if in Figure 1 (a) the node t2 were actually a tag node "c", then there will be a cycle (for more on cycles, see [11]). This paper presents a design and implementation of the new version, which removes all these limitations and

supports arbitrary XML files. In addition, this paper presents the implementation and results of tests on 11 sample XML documents aimed to evaluate the design and implementation.

Contributions. Design, implementation and test results of two versions of the novel online XML compressor, XSAQCT are presented. These two versions are tested and compared with: (1) Send-and-Compress, i.e., sending a single XML file D over the network from node $N1$ to node $N2$ and then compressing *offline* in $N2$; and (2) Compress-and-Send, i.e., compressing D *offline* on $N1$ and sending to $N2$. Recall from [1] that online XSAQCT **not only decompresses** the data whenever enough data is available, but it also *compresses* online, which is essential for the case of a network node $N1$ receiving *streamed* XML data from one or more sources, which are to be stored in a compressed form. The tests show that for existing files online compression is less efficient than offline compression. However, online compression in its natural environment (e.g., streaming) is a more space efficient and faster technique.

This paper is organized as follows. Section II gives a short introduction to the design and functionality of the previous offline version of XSAQCT, and Section III describes its current extension, i.e., online XSAQCT. Section IV is on characteristics of the test suite used in this paper, and Section V provides the description of the implementation and testing results. Section VI describes applications of XSAQCT for online communication, and finally, Section VII provides conclusions and describes future work.

II. OUTLINE OF OFFLINE XSAQCT

For the sake of completeness, we briefly recall here a description of offline XSAQCT; for more details, see [11][7]. Given an XML document D , we perform a single SAX (specifically using Xerces, [17]) traversal of D to encode it, thereby creating an annotated tree $T_{A,D}$, in which all *similar* paths (i.e., paths that are identical, possibly with the exception of the last component, which is the data value) are merged into a single path and each node is annotated with a sequence of integers; see Fig. 1. When the annotated tree is being created, data values are output to the appropriate data containers. Next, $T_{A,D}$ is compressed by writing its annotations to one container and finally all containers are compressed using selected *back-end compressors*, e.g., GZIP [18]. While GZIP was chosen (because HTTP standard uses it), another suitable data compressor can be used as a back-end compressor.

Note that if there was another node labeled “c” in Fig. 2 c) then the document $D2$ would have a cycle.

III. ONLINE XSAQCT

In this section, we present our online algorithms.

A. Notations and Terminology

In this paper, XML documents may have *mixed* contents, assuming “full mixed content”, i.e., there exists a text child separating any two siblings, and there are text children respectively before the first child and after the last child.

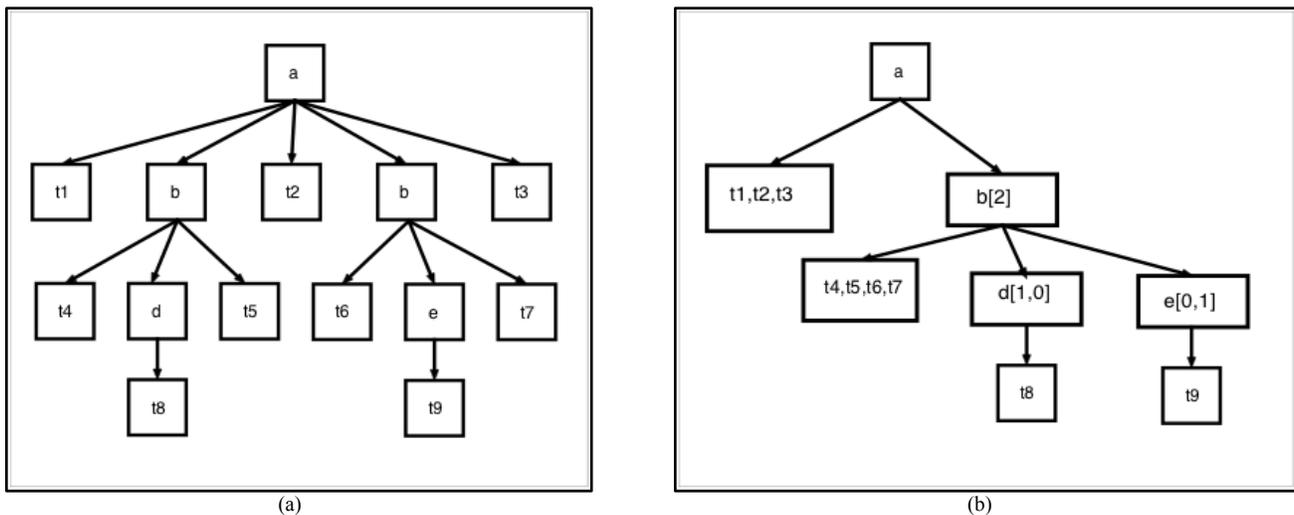


Figure 1. XML document D (a), the annotated tree $T_{A,D}$ (b)

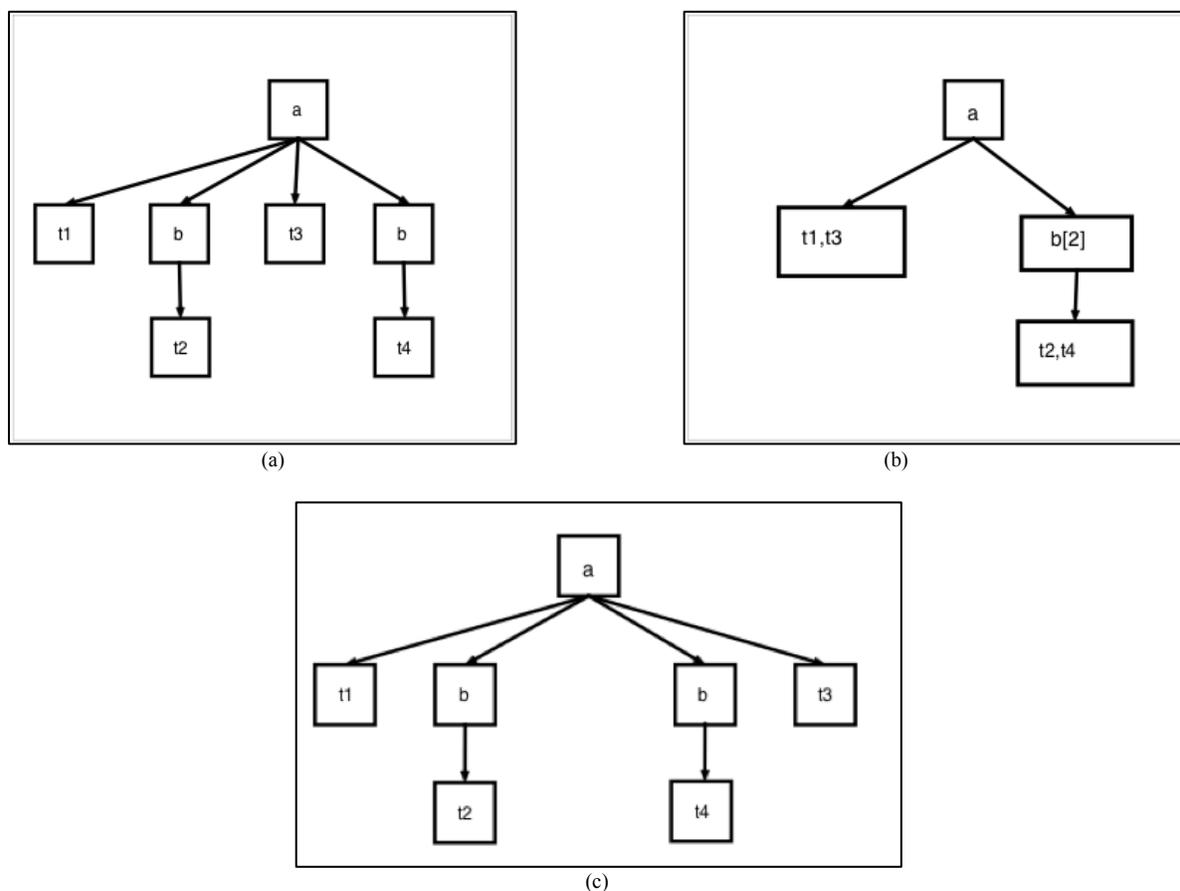


Figure 2. XML document D1 (a), the annotated tree for D1 (b), another XML document D2 (c)

Example of full mixed contents is shown in Fig. 1 (a) and its annotated tree is shown in Fig. 1 (b). The use of full mixed contents is required; otherwise an annotated tree would not uniquely represent every XML document. For example, for the XML document D1 from part (a) and D2 from part (c), the annotated tree shown in part (b) of Fig. 2 is the same. Common occurrences of nodes that do not exhibit the full mixed content property are elements that use font-style tags, e.g.,

`Bold TextOther Text`. Note that to achieve full mixed content, the sending node may have to insert empty text (consisting only of ASCII zero) whenever the text is missing; the receiving end outputting the decompressed file will neglect such empty texts.

In Fig. 2 (c), there would be an empty text between the two occurrences of “b”.

The skeleton tree T_D denotes the tree labeled by tag names (with no annotations) and ANN denotes the sequence of all annotations. Annotations for a node of $T_{A,D}$ may be stored with this node, or the node may store a (logical) pointer to ANN (e.g., the offset within ANN). In the annotated tree, for each node n the text for all similar paths ending with n is stored as the leftmost child of n (strictly speaking a *container* for all texts, separated by ASCII zero); see Fig. 1 (b).

We assume that an annotated tree $T_{A,D}$ is implemented so that following functions are available:

- Node `add_RC(Node n, Tag p, annotation a)` creates and returns a new rightmost child of n with the tag p and the annotation a ;
- void `add_Text(Node n, Text t)` adds text t to the leftmost child of n (creating it if necessary)
- Node `create_Root(Tag p)` creates a new root with tag p ;
- Node `get_LC(Node n)` returns the leftmost child of n ;
- Node `get_RS(Node n)` returns the right sibling of n ;
- bool function `is_Text(Node n)` returns true iff n is a special tree node to store text;
- Node `get_Parent(n)` returns the parent of n ;
- Node `get_Tag(n)` returns the tag of n ;
- Text `get_Text(n)` returns the text child of leaf node n .

In addition, we assume that a data structure Path stores tags or text value, with the operations `append_Node(Path p, Node n)` which appends n to the path p , `append_Text(Path p, Text t)` which appends text T to the path p , `clear_Path(Path p)` which sets the path p to empty, and `set_Path(Path p, int k)` which stores k as the first element of p . Finally, we use the following notations:

- $a(n)$ annotation of the node n
- $a(n)+=j$ increase the last annotation of n by j

$a(n)+="0"$ add “, 0” to the annotation of n,
 e.g., if $a(n)=[1]$ then it becomes $[1,0]$
 $[0^{a(m)},1]$ if $a(m)$ is $[1]$, then $[0^{a(m)},1]$ is
 $[0,1]$ otherwise $[0^{a(m)},1]$ is
 $[0,\dots,0,1]$ where $0^{a(m)}$ is the sum of all
 annotations in $a(m)$, minus 1; e.g., if
 $a(m)=[2,1]$, then $[0^{a(m)},1]$ is $[0,0,1]$.

B. Online Compression

This section describes two algorithms used for online compression, starting with a general description.

SN denotes a sending node and RN denotes a receiving node. SN and RN communicate using message passing; here SN is a producer using *send(packet)*, RN is a consumer using *receive(packet)*, where a packet is defined as a collection of data used for one processing branch (a series of data of the form:[annotation operation, text operation]); finally, synchronization is taken care of by these procedures. SN parses XML and sends packets to RN, which first creates an annotated tree (as described below) and then follows the compression process from XSAQCT [11]. To reduce the overload of sending tag names, the parser creates a dictionary of tags, which is built incrementally by SN and RN. Specifically, for a new tag T, which has not been encountered yet, SN adds T to the dictionary and sends to RN the packet containing the tag and its key in the dictionary. Then, RN uses this packet to update its dictionary, while for an existing packet only the key is sent. As a result, RN can create an annotated tree labeled by indices rather than tags. For the sake of readability the description provided in this paper shows sending and receiving tags rather than indices but our implementation operates on indices.

1. *Basic Algorithm*: The online compression is performed by two procedures, respectively executed by SN and by RN.

```

int k = -1; Path p;
// initially stores only the tag of the root
// of the XML tree
void SN_send_compress(Node n) {
  c = LC(n); // must be text, possibly empty
  append_Path(p, get_Text(c));
  c = RS(c);
  while(c != 0) {
    if(is_Text(c)) {
      append_Path(get_Text(c));
      c = RS(c);
      if(c==0) break;
    }
    append_Node(p, get_Tag(c));
    SN_send_compress(c);
    k++;
    c=RS(c);
  }
  set_Path(p, k);
  send(p);
  clear_Path(p);
  k=0;
} // SN send compress()

```

The pseudo-code for procedure *SN_send_compress()* is shown as if it was a recursive procedure running on the XML tree, but in the actual implementation the tree is not created in memory, instead an event-based SAX [17] parser implements the actions of *SN_send_compress()*. When *SN_send_compress()* is called, it sends a packet of the form (-1, the path of the leftmost path rooted at the root of the tree), and at this time the value of the “current node” c is set to n_k ; then this procedure is called recursively.

```

void RN_receive_compress() {
    bool flag; Node m; Text t;
    Node c; // current node
    receive(k, p1, ..., pN, t);
    if(k==1) { // initialization, the path received starts
        // with a node (the root), create the tree
        c = create_Root(p1);
        add_Text(c, p2);
        for(i=3; i<N; i+=2) {
            c = Add_RC(c, pi, [1]);
            add_Text(c, p(i+1));
        }
    }
    while (true) { // until the final packet
        receive(k, p1, ..., pN, t);
        // the path received starts with a text
        if(k == -2)
            return; // done
        //move current based on the value of c
        for(i=1; i<=k; ++i) // set the current
            c = get_Parent(c);
        add_Text(c, p1);
        //check every tag in the received path
        for(i=2; i<=N; i+=2) {
            flag = false;
            for (m= RS(LC(c)); m <> 0; m= RS(m))
                if(get_Tag(m) == pi) {
                    a(m)+=1;
                    c = m;
                    flag = true;
                    for (every non-text child m of c)
                        a(m) += "0";
                    add_Text(c, pi+1);
                    break;
                } // end of if and of inner for
            if(!flag) {
                c = add_RC(c, pi, [0a(c),1]);
                add_Text(c, pi+1);
            }
        } // for i=1...
    } // while(true)
} // RN_receive_compress()

```

Example 1: Compression.

(a) SN_send_compress()

For the XML file from Fig. 1 (a), we show the *trace* of the execution and packets (numbered p1, p2,...) sent by SN_send_compress(). Packets sent are shown in bold.

```

SN(a): // SN denotes: SN_send_compress
p={a}
c=t1
p={a, t1}
c=b
p={a, t1,b}
SN(b):
c=t4
p={a, t1, b, t4}
c=d
p={a, t1, b, t4, d}
SN(d):
c=t8
p={a, t1, b, t4, d, t8}
c=0
p1: {-1, a, t1, b, t4, d, t8}
p={}
c=t5
p={t5}
c=0
p2: {1, t5}
p={}
c=t2
p={t2}
c=b
p={t2, b}
SN(b):
c=t6
p={t2, b, t6}
c=e
p={t2, b, t6, e}
SN(e):
c=t9
p={t2, b, t6, e, t9}
c=0
p3={1, t2, b, t6, e, t9}
p={}
c=t7
p={t7}
c=0
p4={1, t7}
p={}
c=t3
p={t3}
c=0
p5={1, t3}
// end of trace

```

(b) RN_receive_compress()

Fig. 3 shows the state of the annotated tree after each packet has been processed by RN_receive_compress(), (un-annotated nodes have annotation [1]). Note that the last state shows the same annotated tree as in Fig. 1 (b).

2. *Improved Algorithm:* This algorithm is similar to algorithm 1), but it removes some overhead of sending some packets. According to the *Basic Algorithm*, for Example 2 the packets sent would start with the following packets (\0 is required to denote end of packet):

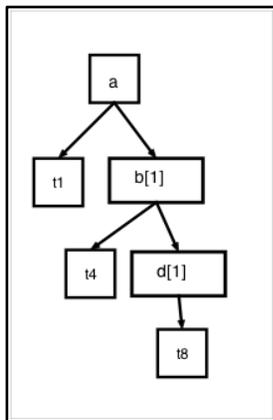
1. {-1, a, t1, b, t2, c, t3, \0}
2. {1, t4, d, t5, \0}
3. {1, t6, e, t7, \0}
4. {1, t8, f, t9, \0}

and the occurrence of consecutive leaf nodes cause at minimum six bytes of overhead with the "1" and "\0" bytes. The *Improved Algorithm* removes that overhead by encoding the packets to be:

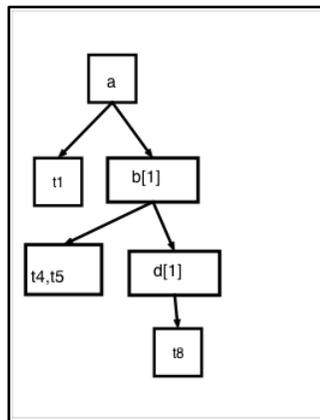
1. {-1, a, t1, b, t2 c, t3, \0}
2. {-2, 3, t4, d, t5 e, t7, t8, f, t9}

where the value of -2 is a special action indicator (similar to what -1 represents in "root node"). One issue not mentioned before is that the packets are encoded in a preorder fashion, implying that the online algorithms have a secondary functionality and through the use of a stack, they can be used to rebuild the original XML file D as opposed to an annotated tree T_D . This is beneficial because it allows a streaming node to pipe XML data directly into a WWW application. Note that there are some boundary cases that need to be considered; for example, consider the following XML fragments:

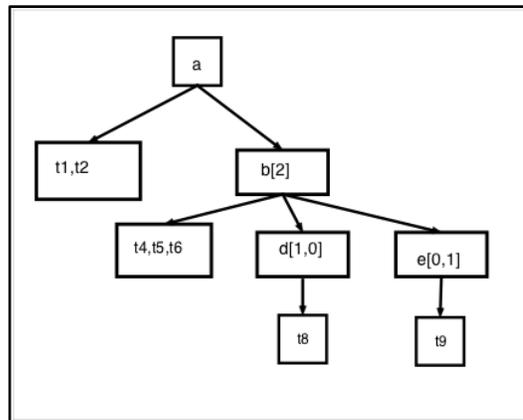
```
<a>
  <b> text </b>
  ...
  <b> text </b>
</a>
```



(a)



(b)



(c)

which are mostly long sequences of leaf nodes. If one parent has say 10,000 such leaf nodes, each with their own text data, then a substantial buffering would be required.

C. Online Decompression

The sending node SN is assumed to be able to decompress all annotations, restore the skeleton tree and send it to RN, then re-annotate it as well as run a procedure SN_send_decompress(AnnotationTreeNode) shown below. As far as the receiving node RN is concerned, it runs a procedure RN_restore_decompress(SkeletonTreeNode) shown below. RN implements the "AA", an abstract data type, which stores sequences of annotations with the following operations (initially, the annotations for every node are un-initialized):

- void AA_delete(Node n) removes the first element of the annotations for n;
- void AA_store(Node n, sequence of integers seq) stores seq as the annotations for n;
- void AA_init(Node n) initializes the annotations for n;
- bool AA_isInit(Node n) returns true iff the annotation for n has been initialized;
- int AA_getFirst(Node n) returns the first element from the annotations for n;
- AA_get_Text(Node n, binary b) where b contains a compressed text, performs the following actions: b is decompressed, stored into a container, and then the iteration AA_nextTextIter(Node n) is started, this iteration returns the next text in the container;
- bool AA_hasReceivedText(Node n) returns true iff the text for n has been received.

Initialization

SN restores the skeleton tree T_D and then the annotated tree $T_{A,D}$ (but it does not decompress text containers), finally it sends the *skeleton* tree to RN, which receives it.

After the initialization, SN runs the procedure SN_send_decompress(AnnotationTreeNode).

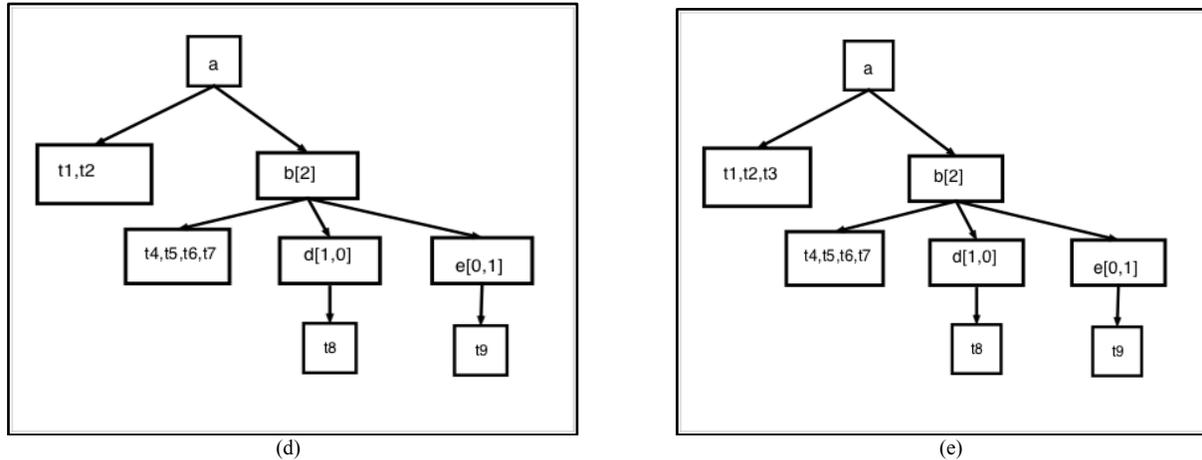


Figure 3. Packets sent by SN_send_compress: (a) p1:{-1, a,t1,b,t4,d,t8}, (b) p2: {1,t5}, (c) p3={1,t2,b,t6,e,t9}, (d) p4={1,t7}. (e) p5={1,t3}

```

SN_send_decompress(AnnotationTreeNode f) {
  for (every child c of f)
    if(isText(c)) send(c, text of c);
    // sends the entire text container,
    else {
      send(ANN(c));
      SN_send_decompress(c);
    }
} // SN_send_decompress()

```

For RN, the following code is executed:

```

output(Document Headings)
output("<" + tag(root of TD) + ">")
RN_restore_decompress(root of TD)
output("</" + tag(root of TD) + ">")
output(Document Trailings)

```

where RN_restore_decompress() is shown below.

Example 2: Decompression.

For the XML file from Fig. 1 (a), Fig. 4 (a) shows its annotated tree and Fig. 4 (b) shows the initial state of the skeleton tree. Table I shows the trace the execution of SN_send_decompress() denoted below as SN() and RN_restore_decompress() denoted below as RN(). T1,...,T4 denote text containers.

IV. CHARACTERISTICS OF THE TEST SUITE

Our experiments used the following 11 files listed here in the order of their sizes (from 5,685.77 GB to 159 KB). Specifically, we use enwiki-latest-stub-articles.xml (from [19]), 1gig.xml (a randomly generated XML file, using xmlgen [20]), enwikibooks-20061201-pages-articles.xml, dblp.xml, SwissProt.xml, enwikinews-20061201-pages-articles.xml, lineitem.xml, shakespeare.xml, uwm.xml (all from the Wratislavia corpus [21]), baseball.xml (from [22]), and macbeth.xml (from [23]).

```

RN_restore_decompress(SkeletonTreeNode f) {
  c = LC(f); //must be text
  if (!AA_hasReceivedText(c)) AA_getText(c);
  Text t = AA_nextTextIter(c); // it shouldn't happen
  // that we reached the end of iteration before this call
  if(!empty_text(t)) output(t);
  c=RS(c);
  while(c<>0) {
    if (!AA_isInit(c)) {
      receive(ann);
      AA_init(c); AA_store(c,ann);
    }
    while (AA_getFirst(c) > 0) {
      output("<" + tag(c) + ">");
      RN_restore_decompress(c);
      a(c)+=-1;
      output("</" + tag(c) + ">");
      output(AA_nextTextIter(LC(f)));
    } // inner while
    AA_delete(c);
    c=RS(c);
  } //outer while
} // RN_restore_decompress()

```

Performance of various algorithms tested in this paper depend on the characteristics of an XML file, such as the size, the number of tags and attributes, the number of unique paths, the distribution of data among the paths and their respective sizes (in Kbytes). Table II provides an overview of these characteristics, where reserved characters are defined as all the static characters defined in the XML grammar (e.g., <, >, /). As it can be seen from Table II, files used for testing greatly vary in various characteristics and in general provide an appropriate test suite. In addition, this suite is designed to simulate streaming, as Send-and-Compress would not be an optimal because it would require *buffering all of the data internally before sending*.

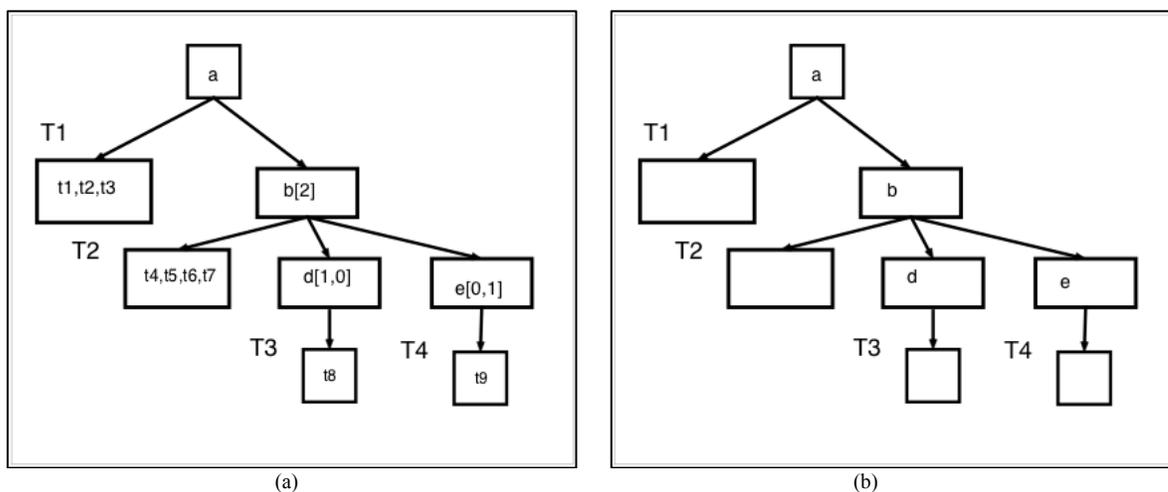


Figure 4.

(a) Annotated tree $T_{A,D}$, (b) skeleton tree T_D

Note that two Wikipedia files (enwiki-books and enwiki-news) have their own schema specifications for rendering to a webpage. However, for the largest Wikipedia XML file (enwiki-latest-stub-articles.xml), for the "text" tag, there is a reference to 112KB of text data, whereas in enwiki-books and enwiki-news that tag would contain all of that data rather than a reference. One common characteristic, not shown in the Table II, is that the height of the XML document, i.e., the length of the longest path from the root of the tree to the leaf, never exceeds six. In other words, XML files used here are often wide but never high, and our design is suited for such files. It should be noted that this is typical of most XML documents used for everyday life and for the Wratlslavia corpus [21], used by most researchers for testing their compressors; however, one can construct atypical XML documents with a large height. Another important characteristic is the number of unique paths in each XML file, which determines how many text containers will be created in the annotated tree. For the test suite used, this number varies from 19 to 548.

Fig. 5 provides a visualization of these characteristics using fractions, e.g., "node tags" represents the percentage of these tags when compared with the entire document (in this figure, some values are too small to be shown). The total sizes of element names and attribute names, calculated as a percentage of the total size, vary respectively from 64.5% to 3.71% and from 8.5% to 0%, and determine how much can be saved using the dictionary for the sender and the receiver. The total sizes of element values and attributes values, i.e., all text values, vary respectively from 93.4% to 10.9% and from 12% to 0%, and determine which data sizes can be reduced, and which cannot. Finally, the total size of reserved characters varies from 20.5% to 3.2%. Fig. 6 shows a comparison of the amount of reducible data, i.e., the amount of overhead through node tags, attribute tags, reserved characters, structure data, etc., in comparison to

the amount of text data (denoted by ELB, our estimated lower bound). The accumulation of text data is defined as the *estimated lower bound* because regardless of the compression scheme applied to the XML structure, this data must be sent to the recipient node. It defines the amount of overhead we are dealing with in comparison to actual data. In general, from this figure and more accurate calculations, one can find out that the ratio of ELB over other reducible data varies from 10% to 89%.

Table III provides sizes of the test files compressed respectively with *offline* XSAQCT and GZIP, compression ratios are calculated as the size of the compressed file over the size of the original file, and finally a comparison of XSAQCT with GZIP is performed by dividing XSAQCT's compression ratio by the GZIP compression ratio (therefore, values less than one indicate that XSAQCT's compression is better). From Table III, it can be seen that in all cases the XSAQCT's compression ratios are better than those for GZIP. The "text" tag in enwiki-latest-stub-articles.xml looks as follows:

```
<page>
<title>Agriculture</title>
<ns>0</ns>
<id>627</id>
<revision>
<id>493785573</id>
<timestamp>2012-0522T06:48:15Z</timestamp>
<contributor>
<username>FrescoBot</username>
<id>9021902</id>
</contributor>
<minor/>
<comment>Bot:[[User:FrescoBot/Section
wikilinks|fixing section wikilinks]]</comment>
<text id="496854391" bytes="112070" />
<sha1>ozdbwwwn9r6if5sz0gcu1558jkr6</sha1>
</revision>
</page>
```

TABLE I. TRACE OF THE EXECUTION OF SN_SEND_DECOMPRESS()

SN(a)	RN(a)	Output
		//initially <a>
c= T1; send(T1)	c=T1; receive(t1,t2,t3), t= t1	t1
c=b[2]; send([2])	c=b, receive[2]; c=b[2]	
SN(b)	RN(b)	
c=T2, send(T2)	f=b; c=T2, receive(t4,t5,t6,t7), t=t4	t4
c=d[1,0], send([1,0])	c=d, receive([1,0]), c=d[1,0]	<d>
SN(d)	RN(d)	
c=T3, send(T3)	c=T3, receive(t8), t=t8, return	t8
	c=d[0,0], c=d[0]	</d>
c=0, return		t5
c=e[0,1], send([0,1])	c=e, receive([0,1]), c=e[0,1]	
SN(e)	c=e[1], c=0	
	c=b[2], c=b[1], c=0, return	
c=T4	c=b[1]	t2
	RN(b)	
	c=T2, t=t6, c=d[0], c=d[], c=e[1]	t6
	RN(e)	<e>
send(T4)	c=T4, receive(T4), t=t9	t9
	c=0, return	
c=0	c=e[0], c=e[]	</e>
Return	c=0, return	t7
Return	c=b[0], c=b[],return	
	return	t3
		//at the end

TABLE II. SOME CHARACTERISTICS OF FILES FROM THE TEST SUITE (SIZES ARE IN BYTES)

File	Node Tags	Attribute Tags	Reserved	Attribute Text	Text Values	Total
enwiki-latest-stub-articles	1,549,965,749	114,862,558	925,621,022	263,920,333	3,011,045,032	5,865,414,694
lgig	185,893,521	25,554,558	92,699,718	46,661,400	815,377,949	1,166,187,146
enwikibooks-20061201-pages-articles	5,791,956	392,912	2,845,193	441,905	146,789,119	156,261,085
Dblp	38,958,602	1,361,043	18,278,604	7,682,331	67,571,145	133,851,725
SwissProt	30,361,262	9,824,703	23,644,591	13,877,139	37,112,515	114,820,210
enwikinews-20061201-pages-articles	3,186,100	221,345	1,485,980	196,858	41,316,971	46,407,254
Lineitem	20,820,560	2	5,114,884	8	6,299,843	32,235,297
Shakespeare	1,808,406	0	898,463	0	4,941,139	7,648,008
Uwm	963,400	24	333,676	72	1,040,357	2,337,529
Baseball	454,720	0	141,530	0	73,032	669,282
Macbeth	40,052	0	19,888	0	103,149	163,089

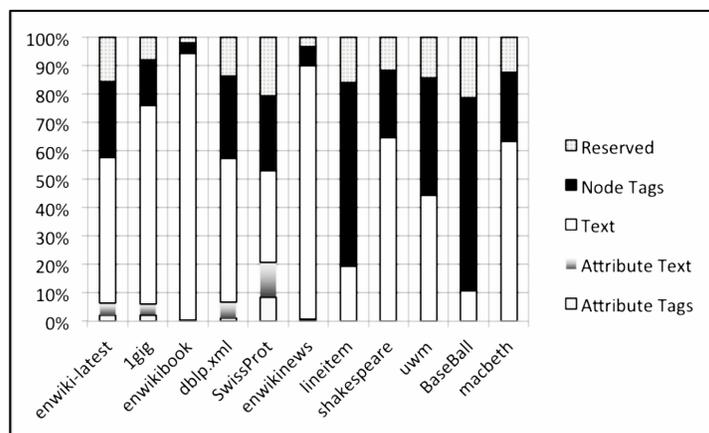


Figure 5. Characteristics of the XML suite

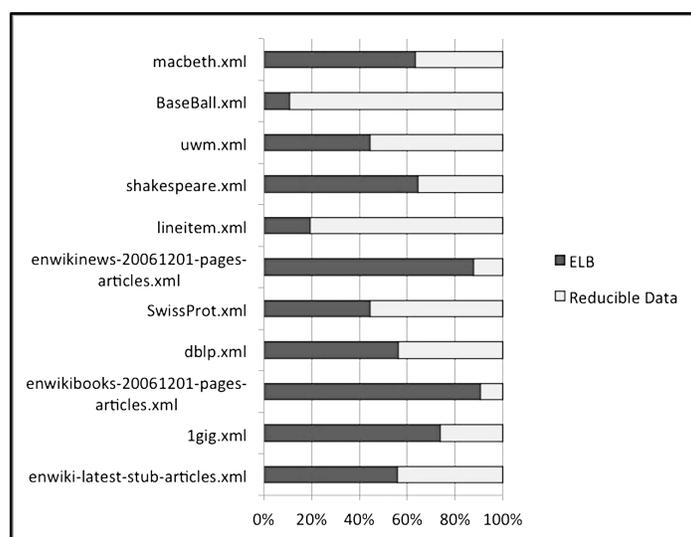


Figure 6. The comparison of reducible and non-reducible data

TABLE III. SIZES (IN MB) AND COMPRESSION RATIOS USING OFFLINE XSAQCT

File	XSAQCT	Compression ratio	GZIP	Compression ratio	XSAQCT compared to GZIP
enwiki-latest-stub-articles	678,268.57	0.1164	931,249.52	0.1599	0.7283
1gig	321,525.55	0.2808	375,695.46	0.3282	0.9743
enwikibooks-20061201-pages-articles	43,475.50	0.2848	44,621.68	0.2923	0.7921
Dblp	18,941.69	0.1449	23,912.73	0.1829	0.5394
SwissProt	7,448.27	0.0664	13,808.91	0.1232	0.9676
enwikinews-20061201-pages-articles	12,322.58	0.2718	12,735.6	0.2809	0.4928
Lineitem	1,401.13	0.0445	2,843.06	0.0903	0.8836
Shakespeare	1,846.92	0.2473	2,090.27	0.2799	0.6298
Uwm	99.44	0.0436	157.90	0.0692	0.6988
Baseball	45.57	0.0694	65.20	0.0994	0.9299
Macbeth	42.37	0.2661	45.56	0.2861	0.8558

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

This section starts with a brief description of the implementation and testing environment, followed by the implementation details. Then, it provides data transfers and timing results of experiments carried out in this environment to evaluate the effectiveness of online compression and decompression.

A. Implementation and Testing Environment

For the implementation language, Java version 1.7.0_05 was used. GZIP [18] was used as the back-end compressor for XASQCT (e.g., compressing the annotation lists and text containers) and in some experiments, for wrapping the sockets I/O stream (to be described later in this paper).

The following three computers were used for testing: (1) an Apple Mac box, here referred to as “SmallMac”, with 2.66 GHz, i7 processor, 8GB 1067 MHz DDR3 RAM, SATA2 SSD; (2) another Apple box, here referred to as “BigMac”, an eight-core with 2.8GHz Quad-Core Intel Xeon chips (Harpertown/Penryn) processors and 12MB of L2 cache per processor; and (3) a Linux box, here referred to as “XPS”, with Intel Duo Core processor, 2.40 GHz, 4GB 1067 MHz DDR3 RAM, and 7200 RPM HDD. The experiments were carried out on LAN using nodes N1 (XPS) and N2 (SmallMac) and N1 and N2 located one hop away. For the sake of completeness note that a 100 Mbit switch connects XPS and SmallMac. Tests were also carried out for sending data from XPS to BigMac and vice versa. The XPS’s upload rate is 150 KB/s and BigMac’s upload rate is 2.5 MB/s. The XPS has a 2.5 MB/s download rate. The routing times between the XPS and BIGMAC, over the Internet (using traceroute) were:

1	dd-wrt : 2949.275 ms 0.290 ms 0.193 ms;
2	modem : 55.319 ms 28.205 ms 19.802 ms;
3	hop 1 : 9.809 ms 9.440 ms 26.398 ms;
4	hop 2 : 9.684 ms 11.549 ms 9.547 ms;
5	firewall : 11.608 ms 11.091 ms 11.307 ms;
6	destination : 11.655 ms 10.395 ms 9.621 ms.

All tag names are encoded as variable sized integers depending on the number of unique elements in our synchronous dictionary. For example, the following approach could be used in determining the encoding:

```
if (elementDictionary.size() < 127)
    //code is a byte
else if (elementDictionary.size() < 32767)
    //code is a short
else // resort to integer
```

Values 127 and 32767 (or $2^{(8-1)}-1$ and $2^{(2*8-1)}-1$) are used because the most significant bit in each encoding is used for handling attribute elements, e.g., if the bit is set, a specific tag includes an attribute added encoding.

For sending annotations, which are possibly very long sequences of non-negative integer values, one modification can be made to the algorithm to improve performance.

There are several possibilities as to how annotation data can be sent from the sending node to the receiving node: (1) sending annotations ANN(n) for each tag node n as this node is encountered during the online decompression; (2) sending the entire sequence ANN of all annotations after decompressing has been completed; and (3) sending ANN compressed (compressing ANN(n) would be useless as these sequences may be short and so the compression may actually be detrimental). For the case of sending all annotations, let us recall from [11] that based on the parents annotation summation, one can figure out the number of integers required for each child, and this is how XSAQCT stores the annotations. It appears that sending compressed annotations should be advantageous and to decide on which option should be chosen, and to test this claim a series of experiments to find out the size of data was carried out. The results are provided in Table IV, in which “uncompressedI” and “uncompressedV” denote respectively sending all data (including annotations) encoded as Integers or Variable Length Integers, and “compressed” means sending all data, including compressed annotations. Based on results from Table IV, compressed annotations encoded as Variable Length Integers (determining the variable length can be stored during the parsing/compressing procedure) are sent on a per-node basis. Finally, note that “Per Node Uncompressed” is not the same as “All Annotations Uncompressed” because of the concept of clean nodes (all annotations are equal to ‘1’) dirty nodes (all remaining nodes). Thus, annotations for clean nodes do not have to be stored; rather nodes are qualified as clean or dirty.

B. Data Transfers and Timing Results

The implementation was tested for offline and online XSAQCT. Four algorithms were compared: (1) Send-and-Compress, denoted below by SC, sending a single XML file D over the network from node N1 to node N2 and then compressing offline in N2; (2) Compress-and-Send, denoted below by CS, compressing D offline on N1 and sending to N2; (3) compressing D using online XSAQCT with the basic algorithm; and (4) compressing D using online XSAQCT with the improved algorithms (both online algorithms were described in Section III B2, in all tables these algorithms are denoted respectively by Online (1) and Online (2)). The sizes of data transferred for each algorithm were computed using both the RAW mode (data sent uncompressed) and the COMPRESS mode (data sent compressed with GZIP).

Each timing test was repeated *three times* and all tables show *the average times* (in seconds) for compression and for decompression, respectively.

There are two possible transmission scenarios: saturated and unsaturated. If the transmission is unsaturated, i.e., the maximum transfer rate is greater than the maximum receiving and then the processing rate, the receiver will never have to block, i.e., wait for data.

TABLE IV. OVERHEAD OF SENDING ANNOTATIONS

File	Per node uncompressedI	Per node compressedI	Per node uncompressedV	Per node compressedV	All annotations uncompressed	All annotations compressed
enwiki-latest-stub-articles	1,126,519,254	691,503,851	794,899,751	691,164,783	1,126,519,147	691,503,851
lgig	373,697,822	328,214,230	337,804,153	327,762,877	373,696,652	328,214,230
enwikibooks-20061201- pages-articles	45,654,608	44,324,089	44,771,053	44,505,494	45,654,524	44,518,913
Dblp	62,291,449	19,433,781	30,420,794	19,218,025	62,291,170	19,396,295
SwissProt	30,485,489	7,663,524	13,175,233	7,469,093	30,484,796	7,627,024
enwikinews-20061201- pages-articles	13,191,594	12,614,494	12,749,145	12,613,409	13,191,510	12,618,319
Lineitem	1,434,856	1,432,365	1,434,860	1,434,880	1,434,791	1,434,759
Shakespeare	2,995,612	1,896,990	2,140,288	1,881,201	2,995,456	1,891,251
Uwm	135,414	101,556	109,031	101,497	298,563	101,826
Baseball	298,620	49,554	100,157	44,296	135,322	46,660
Macbeth	64,452	43,652	47,993	43,575	64,354	43,389

If the transmission is saturated, the receiving node sometimes has to wait for data to process, and so it will sometimes block. Timing is more important for the unsaturated transmission, whereas data transfer is more important for the saturated one. However, the results for latter type of transmission fall in line with what was described in section I.

To test various kinds of environments, we created the three experiments: XPS -> (1) BigMac was heavily saturated; (2) BigMac -> XPS was semi-saturated, and (3) LAN was unsaturated.

In our future work, we will try to develop a saturation metric, e.g., Saturation estimate = amount of time on IO wait

queue / total amount of processing time (the higher the number, the more network-dependent the processing is).

Tables V and VI provide RAW and GZIP data transfer results, respectively. These two tables show that the offline compression CS is always the most space-efficient algorithm, i.e., it transfers the least amount of data. Note, however, that for the GZIP mode the differences between the online algorithms and the offline algorithms are less profound. To explain the reason for these results, note that in a RAW mode, using CS, text and annotations are always compressed, while in online compression the packets (specifically text data) are not compressed.

TABLE V. RAW DATA TRANSFER RESULTS (IN BYTES)

File	File Size	CS	SC	Online (1)	Online (2)
enwiki-latest-stub-articles	5,961,966,106	694,547,020	5,961,966,106	4,124,439,288	4,055,267,064
lgig	1,172,322,551	329,242,185	1,172,322,551	947,901,973	937,671,984
enwikibooks-20061201- pages-articles	156,300,597	44,518,962	156,300,597	143,338,449	143,316,025
dblp	133,862,399	19,396,313	133,862,399	92,448,731	87,775,673
SwissProt	114,820,211	7,627,047	114,820,211	70,294,919	66,572,952
enwikinews-20061201- pages-articles	46,418,850	12,618,367	46,418,850	41,775,533	41,746,933
lineitem	32,235,298	1,434,781	32,235,298	11,294,640	9,609,740
shakespeare	7,647,996	1,891,276	7,647,996	5,774,201	5,617,525
uwm	2,337,523	101,843	2,337,523	1,337,690	1,311,656
baseball	671,924	46,682	671,924	212,536	163,496
macbeth	163,077	43,410	163,077	121,776	118,196

TABLE VI. GZIP DATA TRANSFER RESULTS (IN BYTES)

File	File Size	CS	SC	Online (1)	Online (2)
enwiki-latest-stub-articles	953,599,509	694,547,020	953,599,509	886,753,284	886,449,292
lgig	384,712,148	329,242,185	384,712,148	371,180,220	371,174,799
enwikibooks-20061201-pages-articles	45,692,602	44,518,962	45,692,602	45,070,832	45,064,449
dblp	24,486,638	19,396,313	24,486,638	22,718,443	22,661,594
SwissProt	14,140,327	7,627,047	14,140,327	12,328,793	12,308,808
enwikinews-20061201-pages-articles	13,041,266	12,618,367	13,041,266	12,817,088	12,815,965
lineitem	2,911,297	1,434,781	2,911,297	2,331,926	2,197,236
shakespeare	2,140,436	1,891,276	2,140,436	2,020,343	2,033,234
uwm	161,692	101,843	161,692	142,912	141,733
baseball	66,769	46,682	66,769	54,109	48,225
macbeth	46,658	43,410	46,658	44,450	44,809

Therefore, comparing these ways of compressing data is not quite fair (the difference in amount of data that has to be transferred shows this.) Our future work will consider a way to deal with this issue by not compressing the annotation and text containers in RAW mode.

For all algorithms in GZIP mode, all data for Online (1), Online(2) and SC are compressed. These results are not surprising because offline and online algorithms have several distinctively different features. Specifically, in terms of document scope, online XSAQCT has a scope local to a path and it is forced to interleave more data thereby increasing the

amount of information entropy and reducing the compression ratio. At the same time, offline XSAQCT has a scope of an entire file (and similar data can be compartmentalized by using the container methodology and compressed at a lower rate).

The remaining part of this section discusses timing results. Table VII provides the LAN-based (unsaturated) compression timing results using the RAW mode. For each file, the most efficient timing of the online algorithm is shown in bold face, the most efficient timing of the offline algorithm is shown in italics.

TABLE VII. LAN-BASED RAW COMPRESSION TIMING RESULTS

File	CS	SC	Online (1)	Online (2)
enwiki-latest-stub-articles	<i>431.713</i>	583.283	1514.077	1490.39
lgig	<i>124.715</i>	159.527	335.796	296.67
enwikibooks-20061201-pages-articles	<i>15.774</i>	18.418	32.749	31.478
dblp	<i>12.533</i>	14.136	33.282	32.544
SwissProt	<i>9.415</i>	10.57	31.077	29.705
enwikinews-20061201-pages-articles	<i>4.945</i>	5.084	9.688	9.477
lineitem	<i>2.711</i>	2.923	7.899	7.804
shakespeare	1.708	<i>1.464</i>	2.886	2.539
uwm	0.889	<i>0.367</i>	1.405	1.025
baseball	0.991	<i>0.414</i>	1.276	0.959
macbeth	0.416	<i>0.231</i>	0.428	0.402

TABLE VIII. COMPARISON OF LAN-BASED RAW COMPRESSION TIMING RESULTS

File	SC vs. Online (1)	SC vs. Online (2)	CS vs. Online (1)	CS vs. Online (2)
enwiki-latest-stub-articles	3.5071	3.4523	2.5958	2.5552
lgig	2.6925	2.3788	2.1049	1.8597
enwikibooks-20061201-pages-articles	2.0761	1.9956	1.7781	1.7091
dblp	2.6555	2.5967	2.3544	2.3022
SwissProt	3.3008	3.1551	2.9401	2.8103
enwikinews-20061201-pages-articles	1.9592	1.9165	1.9056	1.8641
lineitem	2.9137	2.8786	2.7024	2.6699
shakespeare	1.6897	1.4865	1.9713	1.7343
uwm	1.5804	1.1530	3.8283	2.7929
baseball	1.2876	0.9677	3.0821	2.3164
macbeth	1.0288	0.9663	1.8528	1.7403

TABLE IX. LAN-BASED GZIP COMPRESSION TIMING RESULTS

File	SC	CS	Online (1)	Online (2)
enwiki-latest-stub-articles	442.281	578.697	1333.298	1322.663
lgig	135.782	159.901	287.905	286.292
enwikibooks-20061201-pages-articles	17.186	18.006	31.93	30.5
dblp	13.159	14.098	31.649	31.092
SwissProt	9.993	10.631	28.565	28
enwikinews-20061201-pages-articles	5.294	5.271	9.938	9.159
lineitem	2.942	2.706	7.714	7.306
shakespeare	1.736	1.38	2.91	2.618
uwm	0.864	0.372	1.383	1.039
baseball	0.998	0.454	1.169	1.07
macbeth	0.42	0.302	0.476	0.388

TABLE X. COMPARISON OF LAN-BASED GZIP COMPRESSION TIMING RESULTS

File	SC vs. Online (1)	SC vs. Online (2)	CS vs. Online (1)	CS vs. Online (2)
enwiki-latest-stub-articles	3.0146	2.9905	2.3040	2.2856
lgig	2.1203	2.1085	1.8005	1.7904
enwikibooks-20061201-pages-articles	1.8579	1.7747	1.7733	1.6939
dblp	2.4051	2.3628	2.2449	2.2054
SwissProt	2.8585	2.7864	2.6870	2.6191
enwikinews-20061201-pages-articles	1.8772	1.7301	1.8854	1.7376
lineitem	2.6220	2.4833	2.8507	2.6999
shakespeare	1.6763	1.5081	2.1087	1.8971
uwm	1.6007	1.2025	3.7177	2.7930
baseball	1.1713	1.0721	2.5749	2.3568
macbeth	1.1333	0.9238	1.5762	1.2848

TABLE XI. COMPARISON OF XPS-BIGMac RAW COMPRESSION TIMING RESULTS

File	CS vs. Online (1)	CS vs. Online (2)	SC vs. Online (1)	SC vs. Online (2)
lgig	2.7629	2.7256	0.8510	0.8395
enwikibooks-20061201-pages-articles	2.9429	3.0194	0.9093	0.9329
dblp	3.8989	3.9489	0.6381	0.6463
SwissProt	7.5003	7.4432	0.6228	0.6180
enwikinews-20061201-pages-articles	3.2949	3.2647	0.9502	0.9415
lineitem	6.0347	5.2047	0.3320	0.2864
shakespeare	2.8017	2.4821	0.8275	0.7331
uwm	5.9046	5.8614	0.4975	0.4939
baseball	1.2254	1.2269	0.4229	0.4234
macbeth	1.5505	1.1284	1.1118	0.8092

TABLE XII. COMPARISON OF XPS-BIGMac GZIP COMPRESSION TIMING RESULTS

File	CS vs. Online (1)	CS vs. Online (2)	SC vs. Online (1)	SC vs. Online (2)
enwiki-latest-stub-articles	1.4705	1.3260	1.1789	1.0630
lgig	1.0699	1.0682	0.9738	0.9723
enwikibooks-20061201-pages-articles	0.9470	0.8740	0.9209	0.8499
dblp	1.0905	1.0019	0.9606	0.8825
SwissProt	1.2103	1.1432	0.7632	0.7209
enwikinews-20061201-pages-articles	0.9099	0.8622	0.8745	0.8286
lineitem	1.3191	1.1031	0.7305	0.6109
shakespeare	0.8870	0.8963	0.8230	0.8316
uwm	0.8489	0.8435	1.1849	1.1773
baseball	0.7132	0.7421	1.5320	1.5940
macbeth	0.8433	0.8762	1.5504	1.6110

TABLE XIII. COMPARISON OF BIGMac-XPS RAW COMPRESSION TIMING RESULTS

File	CS vs. Online (1)	CS vs. Online (2)	SC vs. Online (1)	SC vs. Online (2)
enwiki-latest-stub-articles	1.0844	0.9798	1.1119	1.0046
lgig	1.6375	1.5859	0.8379	0.8115
enwikibooks-20061201-pages-articles	1.8667	1.8947	0.9188	0.9326
dblp	1.8849	1.7962	0.6987	0.6658
SwissProt	2.3250	2.3710	0.6353	0.6479
enwikinews-20061201-pages-articles	1.4902	0.8523	1.5875	0.9080
lineitem	2.1719	2.2143	0.5404	0.5509
shakespeare	1.2205	1.2135	0.7730	0.7685
uwm	0.9653	0.9784	0.9734	0.9867
baseball	1.8118	0.9524	1.8691	0.9826
macbeth	0.7094	0.7117	1.4904	1.4952

TABLE XIV. COMPARISON OF BIGMAC-XPS GZIP COMPRESSION TIMING RESULTS

File	CS vs. Online (1)	CS vs. Online (2)	SC vs. Online (1)	SC vs. Online (2)
enwiki-latest-stub-articles	2.1716	2.0828	2.6257	2.5184
lgig	1.0640	1.0607	0.9467	0.9437
enwikibooks-20061201-pages-articles	0.6199	0.6194	0.9985	0.9976
dblp	0.5847	0.5442	1.1176	1.0403
SwissProt	0.7437	0.7366	1.2010	1.1894
enwikinews-20061201-pages-articles	0.6035	0.5981	0.9672	0.9585
lineitem	1.0138	0.9686	1.3551	1.2947
shakespeare	0.6804	0.6356	0.9356	0.8740
uwm	0.8685	0.8200	0.8685	0.8200
baseball	0.7255	0.7098	0.7255	0.7098
macbeth	0.7083	0.7083	0.7083	0.7083

These results show that for the first seven largest files CS is more efficient than SC, while Online(2) is always more efficient than Online (1). A comparison of timing results for the offline and online algorithms is provided in Table VIII, where “X vs. Y” gives the ratio of the timing result of Y divided by the timing result of X; therefore the value greater than one indicates that X is more efficient than Y. These results indicate that offline algorithms are more efficient than online algorithms.

Table IX is similar to Table VII, but it provides the LAN-based compression timing results using the GZIP mode. These results confirm that for the first seven largest files CS is the most efficient algorithm, while for the remaining four smaller files, SC is the most efficient. A comparison of the four algorithms is provided in Table X, using the same technique as in Table VIII. Results from this table confirm that both offline algorithms are less time-efficient than the online algorithms.

Besides LAN-based tests, two other sets of tests (using the RAW and the GZIP mode) were performed respectively sending data from XPS to BigMac (with 150 KB/s download rate) and sending data from BigMac to XPS (with 2.5 MB/s upload rate). For the former case, the results for the largest (over 5G in size) enwiki-latest-stub-articles.xml file in the RAW mode are not provided because it takes too much time to transfer data. Results provided in Tables XI and XII indicate that for sending data from XPS to BigMac, Online(2) is faster than SC, but the offline algorithm CS is the fastest of the four algorithms. Tables XIII and XIV are similar to Tables XI and XII, but they provide comparison of the timing results for sending data from BigMac to XPS, respectively using the RAW and GZIP mode. Results from these tables are similar to previous results and show that offline algorithms are faster than online algorithms. Now, we describe *decompression*. Here, the **client** rebuilds the XML file and the **server** sends the compressed representation. Therefore, timing results may be disproportionate, because the client has to decompress every text container and then rebuild the document. However, in a client-server paradigm,

where a server may be answering many clients’ requests, in our future work this disproportionality may prove to be more beneficial.

Table XV provides LAN-based decompression timing results using the RAW and GZIP mode. While our algorithms are not very fast comparing to decompression of GZIP-ed file, they always send less data than just using GZIP (see Tables III, V and VI), which is another argument for using XML-based compression techniques. Tables XVI and XVII provide similar results to the results from Table XV, but for sending data between BigMac and XPS (missing row in Table XVII indicates that decompression of the corresponding files was taking too much time). Results from these tables are consistent with other results and show that in non-low-bandwidth situations, online decompression is slow in comparison to just decompressing an ordinary GZIP-ed file. In low bandwidth situations, the added compression proves to overcome the disparity in processing times.

The remaining part of this section describes results of tests aimed to compare both versions of online algorithms with the algorithm referred to as SCU, which performs the offline compression, then it sends the compressed file and finally the receiver performs the offline decompression.

Tables XVIII to XXIII provide respectively RAW and GZIP results of tests. Table XVIII shows that online algorithms are best (with Online(2) being marginally a winner). Note, however, that Online(1) and Online (2) are very similar and “marginally better” falls into the margin of error.

Table XIX shows that base (sending GZIP as is) is the best because data is being sent so fast that there is no point in running an extra algorithm on it (recall from the first paragraph in this paper “low bandwidth networks”).

Table XX shows that SCU is the best, because decompression time is less than time to send online (in raw mode, the amount of data one has to send is very large in comparison to SCU). To understand results from the remaining tables, it is useful to recall the Tables V and VI showing the amount of data to transfer.

TABLE XV. LAN-BASED DECOMPRESSION TIMING RESULTS (IN S)

File	RAW	GZIP	Server	Client
enwiki-latest-stub-articles	594.018	217.179	1542.516	347.794
lgig	114.79	84.579	201.277	185.177
enwikibooks-20061201-pages-articles	14.643	10.741	19.951	11.186
dblp	12.704	6.197	41.191	18.847
SwissProt	9.978	3.453	32.294	7.009
enwikinews-20061201-pages-articles	3.996	2.731	7.284	3.724
lineitem	2.797	1.039	5.947	0.425
shakespeare	0.663	0.51	0.217	0.814
uwm	0.224	0.075	0.548	0.111
baseball	0.07	0.027	0.43	0.184
macbeth	0.048	0.124	0.216	0.07

TABLE XVI. BIGMAC TO XPS -BASED DECOMPRESSION TIMING RESULTS

File	RAW	GZIP	Server	Client
enwiki-latest-stub-articles	2543.355	406.214	1588.146	444.263
lgig	499.474	179.658	260.695	244.751
enwikibooks-20061201-pages-articles	64.419	20.938	26.542	18.478
dblp	55.493	12.37	39.658	18.096
SwissProt	48.8778	9.03	31.338	7.429
enwikinews-20061201-pages-articles	18.875	5.148	8.657	5.37
lineitem	13.044	1.356	6.141	0.768
shakespeare	2.987	1.295	2.333	0.932
uwm	0.702	0.114	0.586	0.141
baseball	0.252	0.084	0.417	0.149
macbeth	0.316	0.069	0.226	0.065

TABLE XVII. XPS TO BIGMAC -BASED DECOMPRESSION TIMING RESULTS

File	RAW	GZIP	Server	Client
lgig	6349.561	2413.013	2076.258	2040.6
enwikibooks-20061201-pages-articles	1027.346	281.652	246.217	232.465
dblp	840.802	147.557	158.461	103.082
SwissProt	730.773	84.544	102.709	40.591
enwikinews-20061201-pages-articles	288.547	78.493	71.056	64.388
lineitem	201.886	15.837	22.677	6.226
shakespeare	47.243	10.156	12.508	8.629
uwm	15.82	0.058	1.785	0.301
baseball	4.672	0.028	0.776	0.187
macbeth	1.131	0.015	0.355	0.073

Tables XXI and XXII show that online algorithms are best, because decompression time is greater than time to send the extra data. Table XXIII shows that for enwiki-latest-stub-articles.xml file, the base algorithm is the best (otherwise, online algorithms are the best). The reason for

this result is that this file is very “text heavy” and for creating packets, text data has to be encoded and buffered. The internal libraries used for this encoding is quite memory/computationally expensive resulting in slow-downs. Our future work will deal with these shortcomings.

TABLE XVIII. LAN-BASED COMPARISON USING SCU OF RAW TIMING RESULTS

File	Online(1)	Online(2)	SCU	Base (sending as is)
enwiki-latest-stub-articles	403.7920	402.9060	2645.9100	594.0180
lgig	115.7440	115.1730	281.7290	114.7900
enwikibooks-20061201-pages-articles	14.1610	14.4710	26.0260	14.6430
dblp	9.7080	8.5060	37.6960	12.7040
SwissProt	8.0270	7.1300	35.2200	9.9780
enwikinews-20061201-pages-articles	4.2470	4.5280	8.7070	3.9960
lineitem	1.9380	1.7380	7.7970	2.7970
shakespeare	1.1030	1.2610	2.8070	0.6630
uwm	0.7790	0.7860	1.1120	0.2240
baseball	0.6660	0.6000	1.4000	0.0700
macbeth	0.3090	0.4060	0.4020	0.0480

TABLE XIX. LAN-BASED COMPARISON USING SCU OF GZIP TIMING RESULTS

File	Online(1)	Online(2)	SCU	BASE (sending GZIP-ed)
enwiki-latest-stub-articles	468.5560	459.3120	2645.9100	217.1790
lgig	118.8480	117.0300	281.7290	84.5790
enwikibooks-20061201-pages-articles	15.2410	15.1560	26.0260	10.7410
dblp	12.3800	12.8280	37.6960	6.1970
SwissProt	10.6750	8.7150	35.2200	3.4530
enwikinews-20061201-pages-articles	4.3790	4.6250	8.7070	2.7310
lineitem	2.7780	2.3230	7.7970	1.0390
shakespeare	1.4140	1.4540	2.8070	0.5100
uwm	0.7640	0.7800	1.1120	0.0750
baseball	0.5810	0.5040	1.4000	0.0270
macbeth	0.3360	0.3240	0.4020	0.1240

TABLE XX. XPS-BIGMAC-BASED COMPARISON USING SCU OF RAW TIMING RESULTS

File	Online(1)	Online(2)	SCU	Base (sending as is)
lgig	5337.4770	5224.8040	2578.4710	6349.5610
dblp	801.7320	789.7230	349.1410	1027.3460
SwissProt	517.5450	480.0680	214.6720	840.8020
enwikinews-20061201-pages-articles	382.7000	362.6890	134.6920	730.7730
lineitem	228.2950	234.8440	99.0010	288.5470
shakespeare	59.9060	50.8520	30.3240	201.8860
uwm	6.7080	5.8030	3.2170	15.8200
baseball	0.6810	0.6200	1.6600	4.6720
macbeth	0.3350	0.2770	0.8030	1.1310

TABLE XXI. XPS-BIGMAC-BASED COMPARISON USING SCU OF GZIP TIMING RESULTS

File	Online(1)	Online(2)	SCU	BASE (sending GZIP-ed)
enwiki-latest-stub-articles	5811.5030	5729.5110	7747.8340	5951.0740
lgig	2186.6430	2169.7950	2578.4710	2413.0130
enwikibooks-20061201-pages-articles	271.0320	246.7336	349.1410	281.6520
dblp	133.3010	125.6450	214.6720	147.5570
SwissProt	69.4780	68.8730	134.6920	84.5440
enwikinews-20061201-pages-articles	70.5520	68.0460	99.0010	78.4930
lineitem	13.1660	10.7800	30.3240	15.8370
shakespeare	10.6080	10.5420	22.3760	10.1560
uwm	1.0240	1.0130	3.2170	0.0580
baseball	0.7710	0.7320	1.6600	0.0280
macbeth	0.3100	0.2950	0.8030	0.0150

TABLE XXII. BIGMAC-XPS-BASED USING SCU COMPARISON OF RAW TIMING RESULTS

File	Online(1)	Online(2)	SCU	Base (sending as is)
enwiki-latest-stub-articles	1703.8838	1691.5070	2710.9930	2543.3550
lgig	381.8510	377.2150	391.7770	499.4740
enwikibooks-20061201-pages-articles	59.0470	57.0220	41.7830	64.4190
dblp	37.9620	36.1100	47.7650	55.4930
SwissProt	28.9790	27.5730	38.4140	48.8778
enwikinews-20061201-pages-articles	17.1140	17.0990	12.6370	18.8750
lineitem	4.6600	3.9560	8.1360	13.0440
shakespeare	2.6050	2.2130	2.9730	2.9870
uwm	0.6210	0.6190	0.9630	0.7020
baseball	0.4240	0.4280	0.8420	0.2520
macbeth	0.2150	0.2140	0.3400	0.3160

TABLE XXIII. BIGMAC-XPS -BASED USING SCU COMPARISON OF GZIP TIMING RESULTS

File	Online(1)	Online(2)	SCU	Base (sending GZIP-ed)
enwiki-latest-stub-articles	692.1700	637.3570	2710.9930	406.2140
lgig	173.8370	167.4110	391.7770	179.6580
enwikibooks-20061201-pages-articles	22.2400	20.8570	41.7830	20.9380
dblp	17.1090	16.2570	47.7650	12.3700
SwissProt	12.1040	11.5370	38.4140	9.0300
enwikinews-20061201-pages-articles	6.3650	5.9300	12.6370	5.1480
lineitem	3.1150	2.7520	8.1360	1.3560
shakespeare	1.8560	1.4790	2.9730	1.2950
uwm	0.6000	0.6030	0.9630	0.1140
baseball	0.4770	0.4440	0.8420	0.0840
macbeth	0.2390	0.2520	0.3400	0.0690

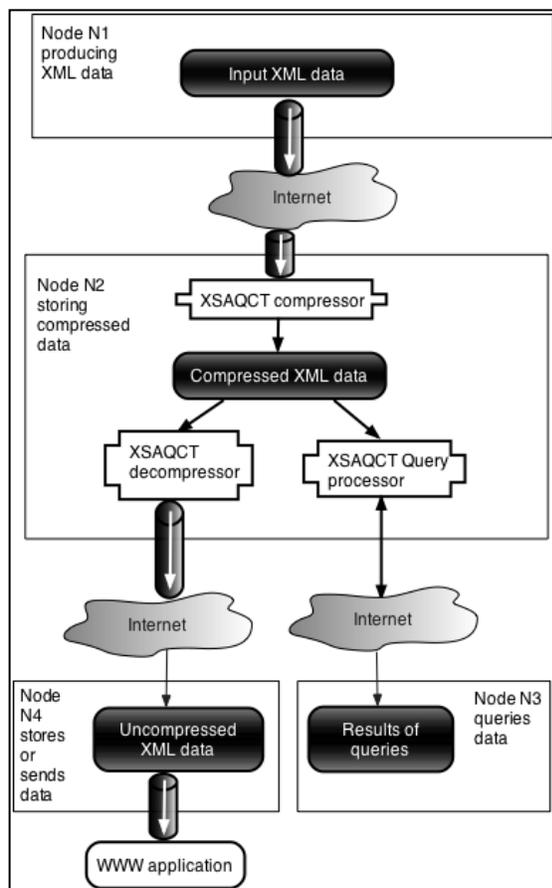


Figure 7. Applications

VI. EXAMPLES OF APPLICATIONS

For the sake of completeness, here we recall from [1] an example of an application. Consider Fig. 7, in which the network node N1 produces XML data to be sent to the network node N2, where they are compressed online by XSAQCT and then they can be queried by N3. This data can also be decompressed online by XSAQCT and sent to a new network node N4, which can either store this uncompressed data, or pipe it into any WWW application.

VII. CONCLUSION AND FUTURE WORK

This paper presented XSAQCT, an online XML compressor/decompressor. The original hypothesis was that the online compression will be more efficient than the offline compression because for the online some actions may be performed "in parallel", i.e., when N1 sends to N2 online, N2 will start decompressing as soon as it gets a chunk of data and at the same time N1 will be sending the next chunk.

The problem with this claim was the dependence on network bandwidth. In low bandwidth situations, several issues might invalidate it because in case of producing data faster than transferring it, all modern operating systems will

intentionally block the process because internal network buffers are full, or cannot accommodate the required data.

This paper provided a brief outline of the implementation and results of tests to evaluate the effectiveness of online XSAQCT; specifically amounts of data transfers and compression and decompression times (in s).

The tests show that for high bandwidth network, and for *existing* files the online compression is less efficient than the offline compression. However, the online compression is superior when compared to offline compression combined with sending the file through the network and subsequent decompression. In addition, the online compression is useful for streaming, i.e., when (potentially generated) XML data is streamed from another network node.

Note that timing results are less important than actual compression ratios because characteristics of the different hardware and operating system may affect timing results as packets are sent through the networking stack.

In our future work, we will attempt a development of a formalization of conditions (which do not factor in processing loads) under which one type of compression would perform better than the other: Let X be the Offline Compression Time, Y be Offline Compressed Size, Z be the Online Compressed Size, and U be the Upload Rate. Assuming that the Online Compression Time is 0 (because there is no waiting period to send data, let $T(\text{Offline}) = (X + (Y/U))$, $T(\text{Online}) = (Z/U)$, and $R = T(\text{Offline}) / T(\text{Online})$. Based on the value of R , one can define (with a pretty high accuracy) the conditions required for online compression to be better than offline compression and vice-versa.

We will also design, implement and test other versions of the online compression by mimicking the SAX parser on the receiving end, rather than sending full information about the nodes (here by mimicking, we mean sending bit-encoded SAX events). Therefore, instead of using a byte, or a variable-length byte encoding, we will investigate working on the bit level.

We will also test different ways of compressing data, and annotations specifically, instead of using ordinary GZIP, we will use BZIP, LZMA, Golomb, and Delta Encoding combined with GZIP. The latter type of compression may be beneficial as typically annotation lists are not a list of random numbers and there is some inherent pattern to them. Instead of using a static dictionary, we will use a more adaptive approach (e.g., a frequency based dictionary) to achieve higher compression rates. Also the future version will add more querying and updating facilities. The complexity of our algorithms will be analyzed, including their footprint.

Finally, we will add parallelization to the online compressor, based on our earlier work reported in [13].

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their helpful comments.

REFERENCES

- [1] T. Müldner, J. K. Miziołek, and C. Fry, "Online Internet Communication Using an XML Compressor," The Seventh International Conference on Internet and Web Applications and Services, ICIW 2012, Stuttgart, Germany, 2012, pp. 131-136.
- [2] H. Liefke and D. Suciuc, "XMill: an efficient compressor for XML data," Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, United States, 2000, pp. 153-164.
- [3] P. Tolani and J. Haritsa, "XGRIND: a query-friendly XML compressor," Proc. of 18th IEEE Intl. Conf. on Data Engineering (ICDE), San Jose, USA, February 2002, pp. 225-234.
- [4] A. Arion, A. Bonifati, G. Costa, S. D'Aguanno, I. Manolescu, and A. Pugliese, "XQueC: pushing queries to compressed XML data," Proceedings of the 29th international conference on Very large data bases - Volume 29, Berlin, Germany, 2003, pp. 1065-1068.
- [5] P. Skibiński and J. Swacha, "Combining efficient XML compression with query processing," Advances in Databases and Information Systems, 2007, pp. 330-342.
- [6] Y. Lin, Y. Zhang, Q. Li, and J. Yang, "Supporting Efficient Query Processing on Compressed XML Files," SAC '05 Proceedings of the 2005 ACM symposium on Applied computing, Santa Fe, New Mexico, pp. 660-665.
- [7] T. Müldner, C. Fry, J. K. Miziołek, and T. Corbin, "Updates of Compressed Dynamic XML Documents," The Eighth International Network Conference (INC2010), Heidelberg, Germany, July 2010, pp. 315-324.
- [8] I. Tatarinov, Z. G. Ives, A. Y. Halevy, and D. S. Weld, "Updating XML," Proceedings of the 2001 ACM SIGMOD international conference on Management of data, Santa Barbara, California, United States, 2001, pp. 413-424.
- [9] S. Sakr, "An Experimental Investigation of XML Compression Tools," CoRR, vol. abs/0806.0075, 2008.
- [10] T. Müldner, G. Leighton, and J. Diamond, "Using XML Compression for WWW Communication," IADIS International Conference WWW/Internet 2005, Lisbon, Portugal, October 2005, pp. 459-466.
- [11] T. Müldner, C. Fry, J. K. Miziołek, and S. Durno, "XSAQCT: XML Queryable Compressor," Balisage: The Markup Conference, Montréal, Canada (August 2009, DOI 10.4242/BalisageVol3.Muldner01 Montréal, Canada, 2009.
- [12] G. Leighton, T. Müldner, and J. Diamond, "TREECHOP: A Tree-based Query-able Compressor for XML," The Ninth Canadian Workshop on Information Theory, Montréal, Canada, June 2005, pp. 115-118.
- [13] T. Müldner, C. Fry, T. Corbin, and J. K. Miziołek, "Parallelization of an XML Data Compressor on Multi-cores," Torun, Poland, 2011, PPAM 2, volume 7204 of Lecture Notes in Computer Science, Springer, (2011), pp. 101-110.
- [14] T. Müldner, J. K. Miziołek, and C. Fry, "Updateable Educational Applications based on Compressed XML Documents," Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011, pp. 369-371.
- [15] W3C, Canonical XML. <http://www.w3.org/TR/xml-c14n>, retrieved on July 20, 2012.
- [16] W3C Extensible Markup Language (XML) 1.0 (Fifth Edition), <http://www.w3.org/TR/REC-xml/>, retrieved on July 20, 2012.
- [17] "Xerces," <http://xerces.apache.org/xerces-j/>, retrieved on July 20, 2012.
- [18] The GZIP home page. <http://www.gzip.org/>, retrieved on July 20, 2012.
- [19] enwiki-latest-stub-articles.xml. <http://dumps.wikimedia.org/enwiki/latest/>, retrieved on July 20, 2012.
- [20] "xmlgen - The Benchmark Data Generator". <http://www.xml-benchmark.org/generator.html>.
- [21] Wratistavia XML Corpus. <http://www.ii.uni.wroc.pl/~extasciitildeinikep/research/Wratistavia>, retrieved on July 20, 2012.
- [22] Baseball.xml. <http://rassyndrome.webs.com/CC/Baseball.xml>. retrieved on July 20, 2012.
- [23] Macbeth.xml. <http://www.ibiblio.org/xml/examples/shakespeare/macbeth.xml>, retrieved on July 20, 2012
- [24] "STAX Parsing. Streaming XML API". <http://java.sun.com/webservices/reference/tutorials/jaxp/html/stax.html#bnbdx>, retrieved on July 20, 2012
- [25] Efficient {XML} Interchange {(EXI)} Format 1.0, <http://www.w3.org/TR/exi/>, retrieved on October 2012
- [26] S. Snyder, "Efficient XML Interchange (EXI) Compression and performance benefits: Development, Implementation and Evaluation Naval Postgraduate School, Monterey, California, Masters Thesis 2010. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA518679>, retrieved on October 2012
- [27] exifcient, <http://exifcient.sourceforge.net>, retrieved on October 2012.

Query-by-Appearance: Visual Query Expansion to Support Domain-Specific Retrieval of e-Books

Shuichi Kurabayashi

Faculty of Environment and Information Studies
Keio University
5322 Endo, Fujisawa, Kanagawa 252-0882, Japan
kurabaya@sfc.keio.ac.jp

Yuka Koike

Faculty of Environment and Information Studies
Keio University
5322 Endo, Fujisawa, Kanagawa 252-0882, Japan
t09334yk@sfc.keio.ac.jp

Abstract—This paper proposes a visual query expansion method called *Query-by-Appearance*, which utilizes techniques developed in the field of editorial design. Our system expands a simple query into one that is informative and well-designed one, allowing users to find their desired e-book intuitively. To simplify the selection of target media data on modern touch screen devices, our system visualizes the retrieved results in a two-dimensional plot that incorporates both layout-based and color-based relevance rankings. We conduct two experiments to evaluate the effectiveness of our Query-by-Appearance method, comparing our automatically generated queries and conventional keyword-based queries in order to determine the extent to which the editorial design templates improve media retrieval. The results show that our visual query expansion method successfully increases retrieval performance for display-oriented media data.

Keywords - e-Book, Search Engine, Editorial Design, Query Expansion

I. INTRODUCTION

This paper describes a *Query-by-Appearance* system [1] and its implementation framework for the EPUB3 e-book data format. Our system utilizes modern editorial design techniques, which enrich the overall appearance of books and magazines, to interpret users' initial queries. By leveraging this editorial design knowledge, we develop an intuitive query editing method to find the desired e-book from the viewpoint of its visual appearance.

Books are traditionally regarded as highly organized media for storing the inherited knowledge of humankind for many years. With the rapid progress of mobile computing technologies, many books are becoming digitized and stored in online libraries and on personal devices. Due to their proliferation, portable and personal devices dominated by a large display, such as tablet computers and smartphones [2],[3], are commonly used to view e-books and web pages. E-books, in particular, are fast becoming popular; the Association of American Publishers reported that total e-book sales in January 2012 constituted 26% of all book sales in the United States [4]. Such proliferation and diversity of digital media increases the demand for an effective retrieval system. By enhancing the retrieval capability for e-books, we enable a wider scope for the sharing of human knowledge and more opportunities for its practical use.

Generic retrieval methods, such as content-based image retrieval systems [5],[6] and keyword-based image search

engines [7], are insufficient for retrieving e-books because users require high domain-specific search quality for frequently accessed media data. For example, a fashion magazine requires a fashion-domain-specific query for its retrieval, whereas very different query is required to retrieve an outdoor and nature magazine. Most approaches to image retrieval on the Internet adopt document search techniques that index images files by associating them with the surrounding text and HTML tag attributes [7]. Due to ambiguity and poor quality, both in the user-supplied keyword query and the textual metadata used to describe the media, web image retrieval systems find it difficult to incorporate visual features of the target media itself in the media retrieval process.

Visual-oriented search mechanisms, which do not use text-based search methods, are promising because users often memorize e-book contents by association with their visual appearance. The overall visual appearance of the page layout is essential to e-book searches because the visual and spatial characteristics of graphical layouts are an important factor in recognizing the contents, as surveyed in [8]. For example, Woodruff et al. [9] indicated that the combination of textual information and thumbnails enhances the usability of Web search engines. However, because conventional book search systems utilize text information, such as tables of contents and indexes [10], there are no studies into layout-based search methods for e-books, in spite of the layout being an important factor for bookbinding.

Thus, this paper proposes a Query-by-Appearance system that improves the query input process by exploiting some knowledge base of book design. Figure 1 illustrates the concept of our Query-by-Appearance system. Query-by-Appearance system adopts editorial design as a template to restructure and modify rough queries entered as input by a user. When the system receives a query consisting of simple lines, it retrieves editorial design templates that are similar to the initial query, and uses them to enrich and improve the query. The system then retrieves actual e-book media data similar to those generated from the query.

This system enables users to find the desired e-book by merely inputting just a simple, intuitive query, because the system ranks e-books according to the similarity of their overall composition, layout, colors, and overview, rather than detailed and trivial differences between them. The system is applicable to the following areas: 1) searching for vaguely remembered visual content, such as the cover of a

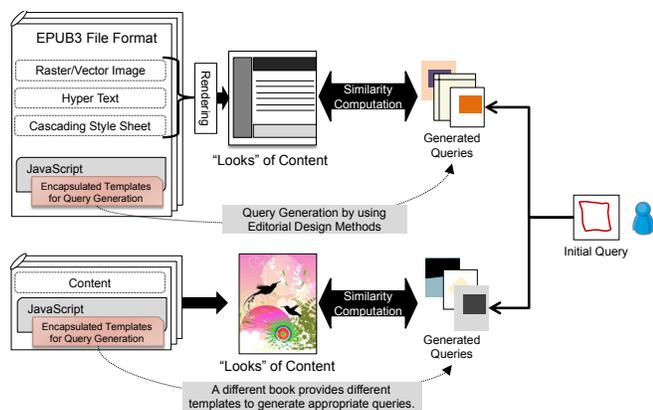


Figure 1. Conceptual view of the Query-by-Appearance System for Style-Oriented e-Book Retrieval Using Encapsulated Editorial Design Templates for Query Generation.

book or a web page, and 2) searching for DVD and CD covers/packaging by preparing appropriate templates.

We implement a prototype system that can be embedded into media data by using JavaScript, which is supported in EPUB3 and HTML5. This prototype system applies a self-search mechanism to the media data according to the internal page images and web page images in the browser history. The system encapsulates the templates derived from the editorial design methods into each e-book according to its style, such as that of a fashion magazine or mystery novel. This design principle makes it possible to interpret user queries by reflecting the characteristics of the target book itself.

The remainder of the paper is structured as follows. In Section II, we discuss several related studies, and Section III introduces several motivating examples to show the advantages of our Query-by-Appearance concepts. Section IV presents an architectural overview of our system. We describe the four fundamental data structures and three core functions of our system in Sections V and VI, respectively. Section VII introduces the prototype system. In Section VIII, we compare conventional keyword-based search method and our visual query-expansion method to show the advantages of editorial design. Finally, in Section IX, we offer our concluding remarks and some ideas for future work.

II. RELATED WORK

In this paper, we present a system architecture that aims to improve the effectiveness of e-book retrieval approaches by exploiting editorial design knowledge as a visual query-expansion source. To cover all of these aspects, we divide our review of related work into three sections: 1) visual similarity analysis of media data, 2) document structure information extraction, and 3) query expansion for e-books.

A. Visual Similarity Analysis of Media Data

Several studies have investigated the development of a search engine that can consider the visual appearance of media. Content-Based Image Retrieval (CBIR) [5],[6] systems have been developed that allow users to search for images by visual similarity. Such systems support Query-by-

Example, which requires a sample image as the query input. However, CBIR methods are unsuitable for finding complex multimedia data such as websites and e-books because they ignore the semantic information associated with the images, such as text and annotations. An alternative method [11] has been proposed that calculates the similarity between the visual components of web pages according to their overall visual appearance. The system proposed in [12] analyzes the visual link structure created by assigning numerical weights to each image. This technique incorporates visual signals into text-based search engines in order to improve the accuracy of conventional searches. Although conventional approaches are effective in retrieving image data when a detailed query, such as a sketch or sample image, is submitted, it is difficult for novices to input such queries, especially in the domain of well-designed e-books. Thus, a new approach to assist users to create domain-specific queries for multimedia data is required.

B. Document Structure Information Extraction

In the field of document database systems, the extraction of document structure information is recognized as an important topic [13]. Recently, Gao et al. [14] proposed a method of analyzing the physical and logical structure of PDF documents, including global typographies, reading order, logical elements, chapter/section hierarchy, and metadata. In a book, the Table of Contents is an important resource for extracting the logical structure [15],[16]. Spatial and semantic information, such as the structure of page numbers, headers, footers, headlines, figures, and body text, can be analyzed to recognize the logical structure of a book. Another method for layout recognition was proposed by Igarashi et al. [17], who used adaptive analysis to determine the implicit structures of card-handling editors. Regardless of these efforts, the development of a general structure analysis method for all document types is nearly impossible, a point highlighted in [18]. It is therefore essential to provide a novel retrieval method that is aware of domain-specific visual features for each type of e-book.

C. Query Expansion for e-books

Traditionally, query expansion is an effective way to help users by modifying their initial query [19]. There is a query expansion method targeting e-books [20] that adopts Wikipedia as the entry point for the book search. This method exploits the link graphs found in Wikipedia to define the relevance metric between books. To apply the query-expansion method to multimedia retrieval, concept-based query expansion [21], which provides high-level suggestions for expanding the original user query, is effective. For example, Hoque et al. [22] applied concept-based query expansion to the retrieval of web images by extracting a list of concepts that are related to the query from Wikipedia. Natsev et al. [23] proposed query expansion methods that integrate text-based keyword relevance analysis and content-based visual example analysis to identify the most relevant visual concepts for a given query. One example of a query expansion method applied to a domain-specific area is that of

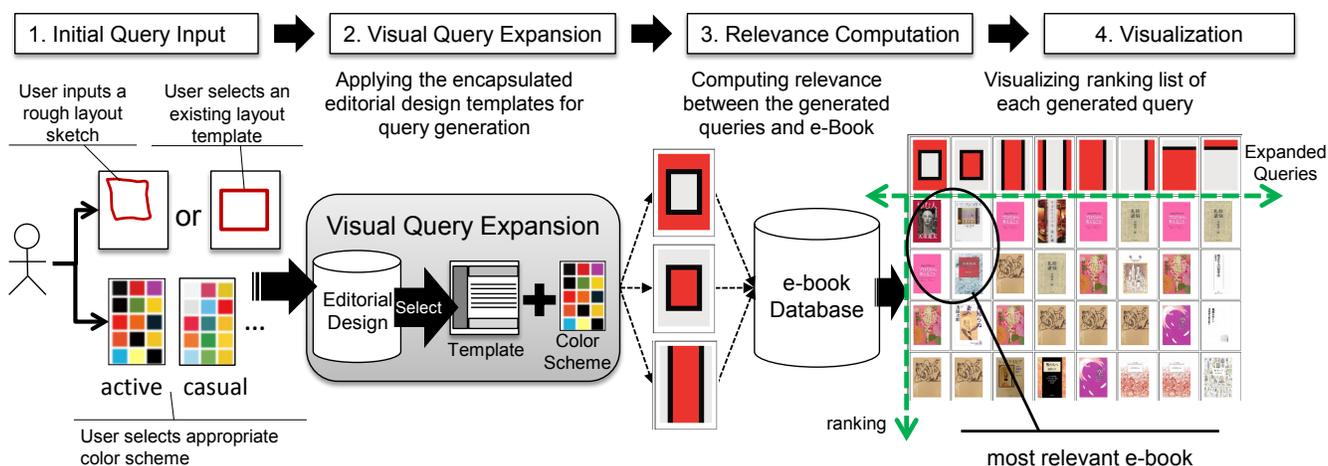


Figure 2. System Architecture of the Query-by-Appearance System.

Joly et al. [24], which describes visual query expansion for detecting specific logos in photographs.

Our Query-by-Appearance system differs from these existing query expansion methods in the following two aspects. First, as shown in Figure 1, Query-by-Appearance provides well-designed examples as alternatives to the initial query of a user. Second, in contrast to keyword-based approaches, Query-by-Appearance first retrieves the relevant editorial design templates, and then uses them to generate various query suggestions. This enables the system to provide well-designed visual queries for retrieving a suitable e-book. Moreover, although previous work on query expansion has mainly focused on the precision of the expanded queries, the main objective of this study is to demonstrate the effect of visual query expansion presentation methods by displaying the retrieval results in a two-dimensional ranking plot that incorporates layout-based and color-based relevance rankings.

III. MOTIVATING EXAMPLE

Our Query-by-Appearance method is used in following situation. For example, a user has 400 books in iPad News Stand. User tries to find book which title, author, published date is uncertain, but remember cover image vaguely. In this case, by using our system, user could retrieve desired book thorough inputting rough visual appearance. When inputting layout and color, it does not have to be correct and specific, because our system generates inputted query into nearest template. On the other hand, our system is not suitable for retrieving books which title and author is clear. In such case, user prefers to use keyword-based search method implemented in the conventional e-book stores and readers.

Another example is dedicated for a user experience aspect. A user draws a rough sketch on a touchscreen of smart devices, such as iPad and Android, according to his/her ambiguous memory. Then the system generates the well-designed e-book image data to retrieve search results. This approach is highly effective in web-based touch screen interfaces because it simplifies the query-input task. The touch screen interface visualizes both the layout similarity

and color similarity in a two-dimensional matrix, enabling users to retrieve the desired e-book according to their preferences (i.e., the weights of the layout similarity and color similarity) by selecting a sub-matrix in the visualized two-dimensional matrix.

IV. SYSTEM ARCHITECTURE

Figure 2 illustrates the architectural overview of the proposed system. Query-by-Appearance provides an indirect e-book retrieval mechanism that is leveraged by a knowledge base of editorial design. Here “indirect” means that the system selects the relevant templates for query generation according to an initial query which consists of rough sketch and several colors. And then, the system generates actual queries by using the selected templates, which are well-designed and have coordinated colors. The system retrieves e-book images that are similar to the generated queries inspired by the initial rough query, and visualizes the results in a two-dimensional ranking. This query-assistance function enables users to input a layout and color scheme by using a web-based touch interface, allowing the search to be conducted intuitively through rough visual appearances, rather than technical knowledge.

The query templates consist of layouts and color schemes. The layout can be specified by a rough sketch or by selecting an appropriate template, and a well-designed color scheme, derived from a color-design book [25], is provided.

The most important function of our system is the visual query expansion function that utilizes both layout and color to enrich and extend the input query. This function employs a design template database that stores layouts and color scheme data. The templates are defined by sets of matrices; thus, the system is able to select relevant templates by calculating the similarity in the layout structure. In addition, 20 color schemes are provided by the combination of 102 colors.

The Query-by-Appearance system for e-books performs the following six steps:

1. The user inputs a rough layout sketch on the HTML5 Canvas system.

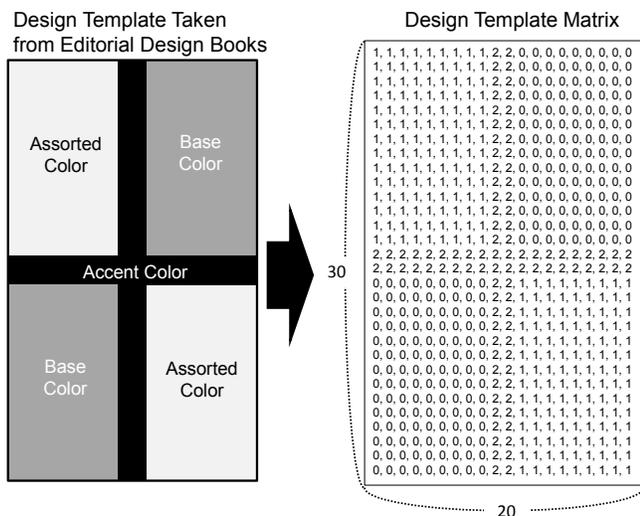


Figure 3. Structure of the Design Template Matrix.

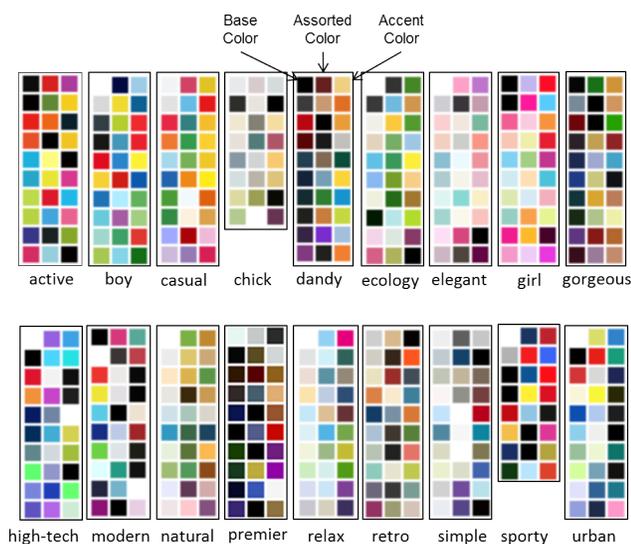


Figure 4. Color Schemes taken from an Editorial Design Book [25].

2. The system converts the rough layout query input into a matrix, which it compares with 30 templates stored in the knowledge base.
3. The system generates colored templates by adding color to the selected layout using the 20 color schemes in the knowledge base. The sets of colored templates are automatically expanded to image matrices that include the layout structure and the color scheme.
4. The system calculates the similarity between the expanded query matrices and e-book image data stored in an e-book database. The e-book image data is preliminarily clustered into 102 colors.
5. E-book cover images that contain similar editorial design characteristics are visualized in a two-dimensional ranking plot with a generated query axis and a ranking axis.

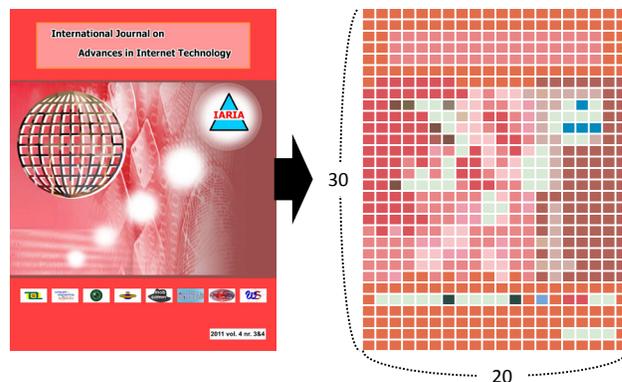


Figure 5. e-Book Matrix Generation.

6. Further retrieval is conducted when the user selects a color on a sub-matrix in the principal two-dimensional matrix.

V. DATA STRUCTURES

This section shows data structures used in our visual query expansion method. Our system contains four fundamental data structures: A) Query Template Knowledge Base from Editorial Design, B) Query Matrix, C) e-book Database, and D) Visualization.

A. Query Template Knowledge Base from Editorial Design

Our editorial design knowledge base for visual query expansion consists of the layout template, and the color scheme. The left side of Figure 3 shows an example layout. The layout template matrix is the set of matrices. The right side of Figure 3 shows an example of a design template matrix. Layout information consists of a base color, an assorted color, and an accent color, which are assigned the numerical identifiers 0, 1, and 2, respectively. Each of color types defines the role of a color. the detailed and formal definition of those three colors are given later in this section. The layout template matrix T_L is defined as follows:

$$T_L := \begin{bmatrix} t_{[0,0]} & \dots & t_{[0,m]} \\ \vdots & \ddots & \vdots \\ t_{[n,0]} & \dots & t_{[n,m]} \end{bmatrix} \quad (1)$$

where $t_{[i,j]}$ indicates the color type at location $[i, j]$ in the template matrix. The system colors each column according to this number. It is important to mention that our Query-by-Appearance model does not assign specific colors to the layout matrix statically, but provides a dynamic color assignment mechanism to generate the query according to user demands. Thus, the layout template T_L only specifies the role of each color as the base color, assorted color, or accent color. According to these roles, the system dynamically assigns the actual colors defined in the color scheme template.

The color schemes are combinations of colors suitable for joint use. Figure 4 shows several examples of color schemas taken from an editorial design book [25]. Each color scheme consists of a base color, an assorted color, and an accent color. We have assigned an adjective to these color schemes in order to group them; the adjectives represent human color



Figure 6. Screenshot of the Prototype System for Visualizing e-book Search Results. In this screenshot, a user selected the layout template.

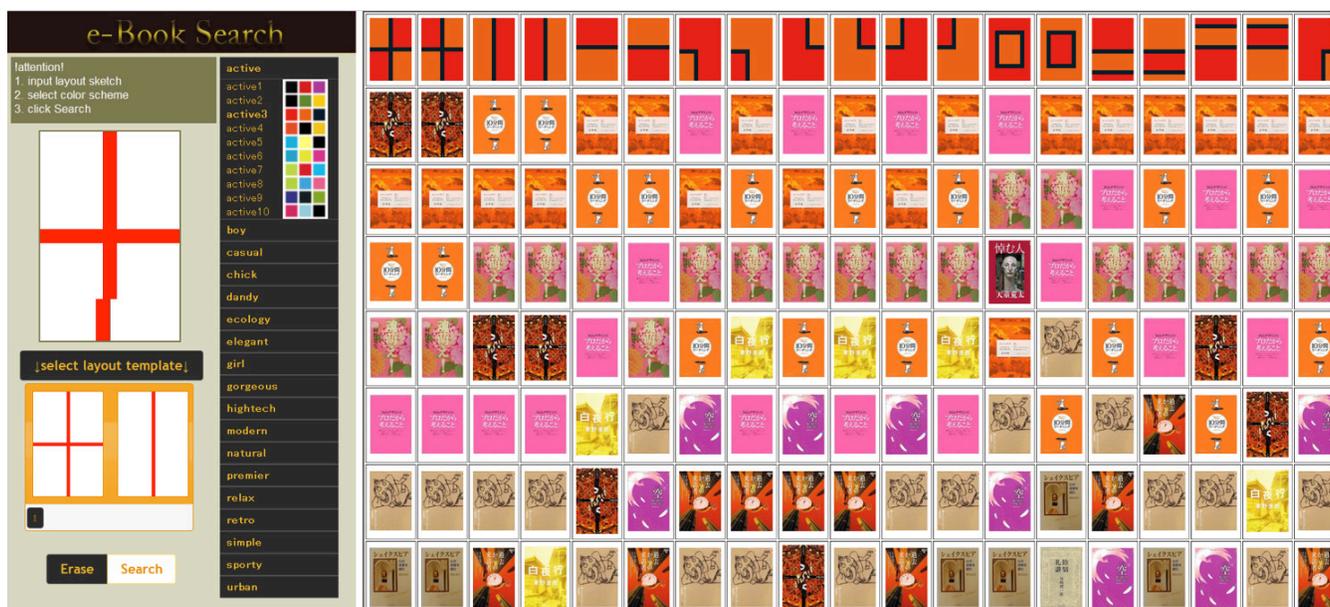


Figure 7. Screenshot of the Prototype System for Visualizing e-book Search Results. In this screenshot, a user sketched the layout to retrieve design templates.

perceptions. In total, we have defined 18 groups of color schemes by using a color scheme definition dictionary [25].

The color scheme is a triplet of colors, which is a combination of 102 primitive colors. As shown in Figure 4, each color scheme is assigned a specific adjective. Our system draws colors in each column according to this scheme, thus expanding the queries in order to generate e-book image data. The color scheme CS is defined as follows:

$$CS := \langle C_{base}, C_{assorted}, C_{accent} \rangle$$

where C_{base} , $C_{assorted}$, and C_{accent} denotes colors modeled in HSV color space. Our model represents colors using the HSV (hue, saturation, and value) color space. HSV is widely adopted in image and video retrieval because it describes perceptual and scalable color relationships. In order to select

appropriate colors, we have selected 102 Munsell basic colors [26]. The details of those colors are as follows:

- A base color is the dominant color in the layout. It is used as the background or main theme color. This color occupies over 70% of the surface.
- An assorted color is the secondary dominant color. It has strong relationship with the base color. This color supports and enhances the impression of the base color. The assorted color occupies 25% of the surface.
- An accent color is a salient color that occupies less than 5% of the surface. It determines the structure and texture of the image. This color is very important because it is recognized as forming the basic shape of a figure drawn on the material.

B. Query Matrix

Our system converts each query into a matrix that represents the rough input of the user, such as simple lines drawn on HTML5 Canvas. We call this the query matrix, which is important for calculating the similarity between user input and layout templates in the knowledge base. The query matrix Q is defined as follows:

$$Q := \begin{bmatrix} w_{[0,0]} & \cdots & w_{[0,m]} \\ \vdots & \ddots & \vdots \\ w_{[n,0]} & \cdots & w_{[n,m]} \end{bmatrix} \quad (2)$$

where $w_{[i,j]}$ indicates whether a line is present at $[i, j]$ in the query. The system models an initial query in pixel level granularity because the templates are designed in a raster format. This is reasonable for dealing with many editorial design templates utilizing various design techniques such as blur effect and shadowing effect. Weights are assigned to each column as follows: if a line passes through a column, we assign a weight of 1.0 to the column; regions surrounding columns with a weight of 1.0 are assigned a weight of 0.5; if no line passes through a column, we assign a weight of 0.0. These weights are used to find the inner product of the user query matrix and the layout template matrix defined in the previous subsection.

C. e-Book Database

Our system converts JPEG-format image data into a matrix structure by clustering the image into the 102 system colors. We call the resultant data structure an e-book matrix. This is a 30×20 data matrix in which each cell represents the color at the corresponding point in the image, as shown in Figure 5. The e-book matrix is used to calculate the similarity with the expanded query matrix obtained from the user input and template knowledge base. The e-book database matrix D is defined as follows:

$$D := \begin{bmatrix} d_{[0,0]} & \cdots & d_{[0,m]} \\ \vdots & \ddots & \vdots \\ d_{[n,0]} & \cdots & d_{[n,m]} \end{bmatrix} \quad (3)$$

where $d_{[i,j]}$ represents each element of the matrix, and $[i, j]$ represent the rows and columns, respectively, of the image data in the matrix.

VI. CORE FUNCTIONS

Query-by-Appearance system consist of three core functions as follows: A) the visual query expansion function, B) the query-by-appearance function, and C) the visualization function. When the system receives an initial query, which is a rough sketch, the system applies the visual query expansion function to generate several queries by making the initial query well-formed according to the editorial design templates. And then the system applies the query-by-appearance function to calculate similarity scores for each generated query and each e-book in the database. Finally the system visualizes the retrieval result by ranking e-books in the descending order of scores.

A. Visual Query Expansion

Step-1: Selecting the Most Relevant Template. The system selects the most relevant template by comparing the

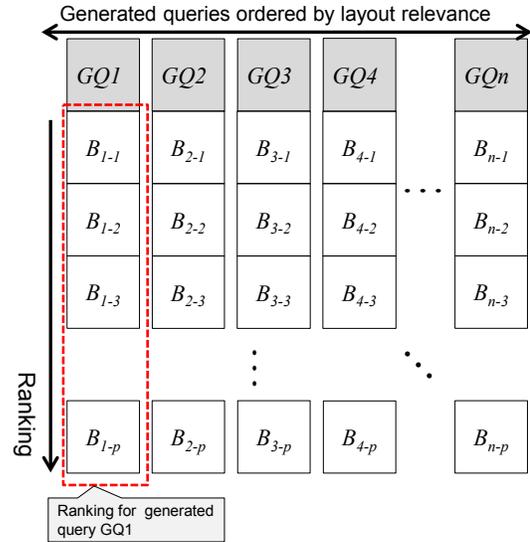


Figure 8. Visualization Matrix Structure.

user input with the template matrices stored in the template knowledge base. The system uses the selected templates to generate actual queries. We call this “query assistance function”. The inner product of the transformed user input matrix, which consists of sketched line data, and the template matrices stored in the template knowledge base is calculated in order to analyze the degree of similarity. Higher values of the inner product indicate greater similarity.

The function is defined as follows:

$$f_{layout}(q, t) := \sum_{i=1}^w \sum_{j=1}^h q_{[i,j]} t_{[i,j]} \quad (4)$$

where $q_{[i,j]}$ indicates whether a line is present at position $[i, j]$ in the query matrix, and $t_{[i,j]}$ indicates whether a line is present at $[i, j]$ in the template matrix. This calculation measures the line similarity between the query matrix and the template matrix by checking all the pixels. This similarity measure is used for only choosing the relevant template, and is not used for actual e-book retrieval phase.

Step-2: Assigning Colors to the Template. Our system provides two strategies for assigning colors to a template. The first strategy allocates a base color to the dominant area of the template and allocates an assorted color to the secondary dominant area of the template. This strategy is called “straight color assignment”. The straight color assignment strategy keeps original roles of the base color and the assorted color. In contrast, the second strategy allocates a base color to the secondary dominant area of the template and allocates an assorted color to the primary dominant area of the template. This exchange of the base color the assorted color is a well-known technique for creating variations of color scheme. The color assignment function is defined as follows:

$$f_{assign}(T_L, CS) \rightarrow Q := \begin{bmatrix} hsv_{[0,0]} & \cdots & hsv_{[0,m]} \\ \vdots & \ddots & \vdots \\ hsv_{[n,0]} & \cdots & hsv_{[n,m]} \end{bmatrix} \quad (5)$$

where T_L indicates the template data, and CS denotes a color scheme containing three HSV colors.

Step-3: Color Space Conversion. The system converts RGB color values of each image to HSV at the pixel level. We adopt the following well-known RGB to HSV conversion equation.

$$V = \max(R, G, B) \quad (6)$$

$$S = 255 \times \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \quad (7)$$

$$H = \begin{cases} 60 \frac{B - G}{\max(R, G, B) - \min(R, G, B)} & R == \max(R, G, B) \\ 60 \left(2 + \frac{R - B}{\max(R, G, B) - \min(R, G, B)} \right) & G == \max(R, G, B) \\ 60 \left(4 + \frac{G - R}{\max(R, G, B) - \min(R, G, B)} \right) & B == \max(R, G, B) \end{cases} \quad (8)$$

B. Query-by-Appearance

Our system calculates the correlation between the generated query matrix and e-book image data matrices in order to retrieve the relevant e-book image. The Query-by-Appearance operation is defined as follows:

$$f_{qba}(Q, B) := \sum_{i=1}^w \sum_{j=1}^h \Delta_{godlove}(Q_{[i,j]}, B_{[i,j]}), \quad (9)$$

$$\Delta_{godlove} := \frac{2S_1S_2 \left(1 - \cos \left(2\pi \frac{|H_1 - H_2|}{100} \right) \right) + (|S_1 - S_2|)^2 + (4|V_1 - V_2|)^2}{2} \quad (10)$$

where $Q_{[i,j]}$ denotes a color at the specific point in the generated query matrix, and consists of HSV components H_1 , S_1 , and V_1 . $B_{[i,j]}$ denotes a color at a specific point in the e-book image data, and consists of H_2 , S_2 , and V_2 . This calculation measures the color similarity between the expanded queries and the e-book image data for each block. We employ Godlove's delta equation [27] to calculate the distance between two HSV colors. To improve the precision of color distance, the CIEDE2000 delta equation [28] can be applied. We adopt Godlove's delta equation here due to its faster computational performance.

Our system uses a two-dimensional ranking mechanism to present the results of the e-book image search. This visualization format enables users to select books on the basis of their layout and color similarity. The system samples template data from a wider field than the user's input in order to display the retrieval results interactively on the basis of the layout and color choices.

The two-dimensional ranking procedure employs the following steps: 1) The generated queries are shown on the horizontal axis. Each column contains a particular query template and color. 2) The ranking of e-book image data is shown on the vertical axis. 3) The system stacks the e-book image data according to the similarity score of queries and e-book metadata in each cell.

C. Visualization

The system visualizes the layout and color similarity using a two-dimensional ranking, as shown in Figure 6 and Figure 7. Results with similar editorial design characteristics are visualized in a grid layout, where the horizontal axis corresponds to the variation of expanded queries and the vertical axis corresponds to the relevance ranking.

The visualization model places the most relevant e-book in the upper-left position because this placement implies a high relevance scores for both the layout and color scheme. This ranking visualization scheme consists of two relevance scores for each cell. The ranking array $Cell_x$ is defined as follows:

$$Cell_x := \langle L_i, C_j, \{B_{x-1} \cdots B_{x-p}\} \rangle \quad (11)$$

where L_i is the layout ID, C_j is the color ID, and $\{B_{x-1} \cdots B_{x-p}\}$ determine the ranking order in a column. The order of books in a certain column depends on their color relevance. Figure 8 shows the basic layout of this visualization model. Figure 6 and Figure 7 show screenshots of prototype system implementation that visualize the above model. In this model, each cell represents a specific "rank" of the corresponding e-book. Our system generates the ranking according to the relevance score, which is calculated with inner product, by sorting them in a descending order.

VII. SYSTEM IMPLEMENTATION

We have implemented a prototype of the Query-by-Appearance system that calculates the similarity between generated queries and editorial design templates. The screenshots of the prototype, which uses HTML5 Canvas and JavaScript, is shown in Figure 6 and Figure 7. In Figure 6, a user selected the layout template, and the system generated the query according to the selected layout and color schemes. In Figure 7, a user sketched the layout to retrieve design templates, and the system listed the ranking of the layout templates and assigned the colors for the retrieval results. An important feature of this implementation is that it uses standard web technologies, which are also used by the current EPUB3 specification. Thus, we can apply our method to EPUB3-based e-books without significant modification.

Figure 9 shows the detailed architecture of our prototype system, which specifically includes the modern HTML5 technologies Canvas API, Web Storage API, and Web Workers API. The system consists of the following three modules: a query editor, a visual query expansion system, and a search result visualization engine. We now describe these components in detail.

The main user interface is the query editor, which uses HTML5 Canvas to produce two-dimensional interactive vector graphics. Users sketch the layout query on the query editor, and select a color scheme. When they have finished drawing, the query editor encodes the figure into JavaScript Object Notation (JSON) format and passes it to the visual query expansion system. This procedure allows our system to share the JSON-encoded figure among multiple web workers to parallelize the execution of visual query expansion.

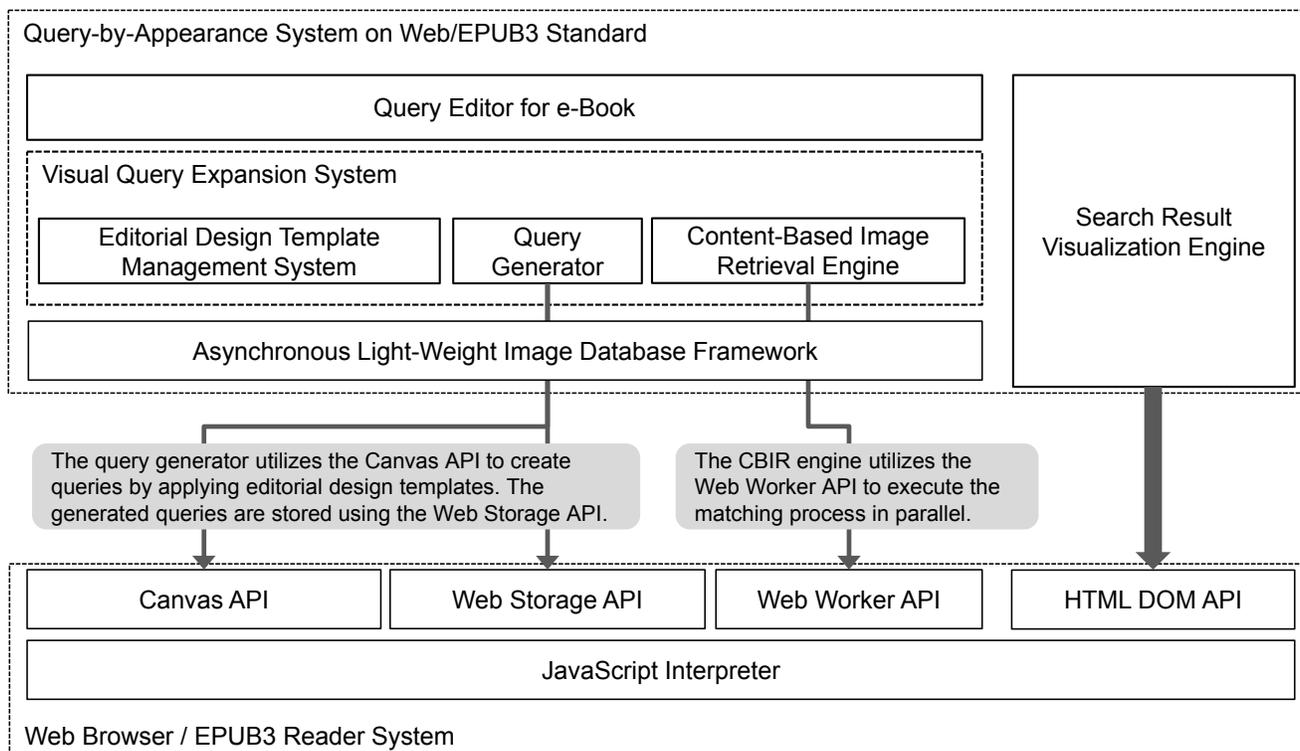


Figure 9. Prototype System Architecture of the Query-by-Appearance System Implemented Using Modern HTML5 Technologies.

The visual query expansion system is a core component of the proposed system. It expands the input query into well-designed layout and color information by retrieving relevant editorial design templates and assigning colors to the retrieved templates. The visual query expansion system consists of the following sub-modules: an editorial design template management system, a query generator, and a content-based image retrieval engine. The editorial design templates are embedded in an HTML file as plain text, and the editorial design template management system parses this file to reconstruct the editorial design template objects. By storing them in HTML5, we can deploy various templates according to the e-book content. The query generator applies the retrieved templates to generate actual queries, and stores the generated queries using the Web Storage API in order to reduce the communication overhead between web browsers and the server. Finally, the content-based image retrieval engine executes the generated queries over the database contents. This retrieval process is parallelized by the Web Workers API, and the retrieved e-books are presented to the user by the search result visualization engine.

The most important feature of this architecture is that the query expansion process and the data retrieval process can be executed on the client side. This client-side execution framework is beneficial as it permits our method to retrieve contents that are inside a book. In addition, this system spawns real OS-level threads from the Web Workers API to parallelize the e-book retrieval process. Modern HTML5 technologies enable us to implement complex processes in web browsers.

VIII. EVALUATION

This section evaluates the effectiveness of our Query-by-Appearance system when applied to the existing books. The experimental data comprise 130 book cover images from Amazon.co.jp. We perform two evaluation experiments: Experiment 1 is an evaluation of retrieval precision, and Experiment 2 is an evaluation of the effectiveness of our system by measuring the time consumed for retrieving the desired book by using our system and the conventional book search system.

A. Experiment-1: Outline of Experimental Studies

In this section, we evaluate the effectiveness of our system by examining its retrieval precision. The aim of this experiment is to clarify the effectiveness of retrieving book cover images by utilizing the editorial design methodology. We compare the retrieval precision using two search methods: 1) queries are generated using only layout templates and 2) queries are generated by integrating layout templates and color scheme templates. We show that the integration of a layout template and a color template makes a significant contribution to the e-book retrieval result.

For this experiment, we prepared: 1) 130 book cover images from Amazon.co.jp, 2) five queries, and 3) five answer data sets for each query. The queries and answer data sets were formed in consideration of the basic structures of editorial design: a symmetrical layout, a diagonal layout, and a layout with a gravity point in the center. Thus, the following five queries were chosen (see Figure 10):

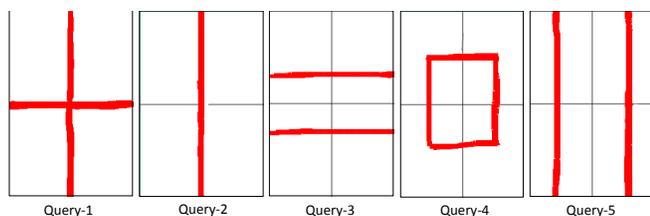


Figure 10. Five Experimental Queries drawn on the Canvas.



Figure 11. Top 20 Retrieved Images for the Combination of Query 3 and "cool" (left), and Query 1 and "sporty" (right).

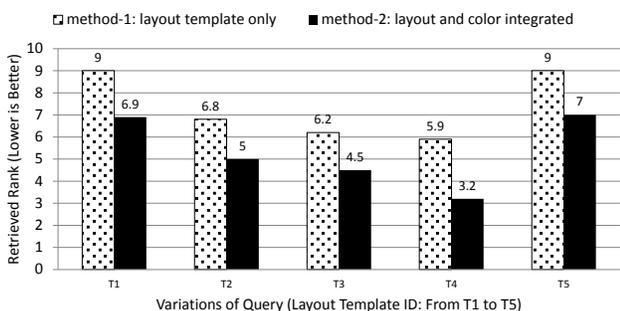


Figure 12. Retrieved Rank Scores Using Search Method 1 (left-hand bars) and Search Method 2 (right-hand bars).

- Query-1. Draw a cross shape to divide the canvas into four sections.
- Query-2. Draw a vertical line along the center of the canvas to divide it into two sections.
- Query-3. Draw two horizontal lines on the canvas to divide it into three sections.
- Query-4. Draw a rectangle in the center of the canvas to divide it into two sections.
- Query-5. Draw two vertical lines on the canvas to divide it into three sections.

We have used color schemes that consist of three colors (as shown in Figure 3 and 4). The system calculates the similarity between the queries and the generated e-book image data.

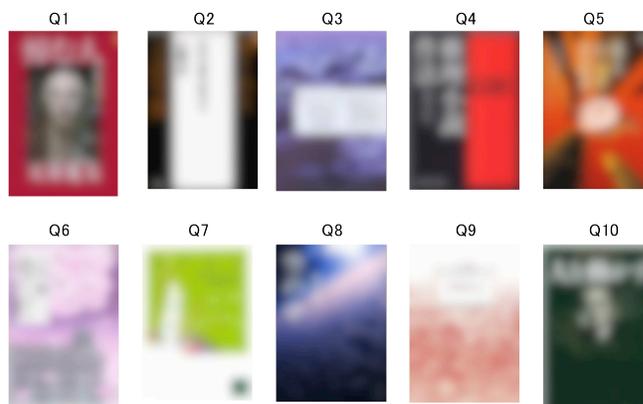


Figure 13. Ten Books to be Retrieved in Experiment-2 (Images are Shaded Off for Protecting Copyrights).

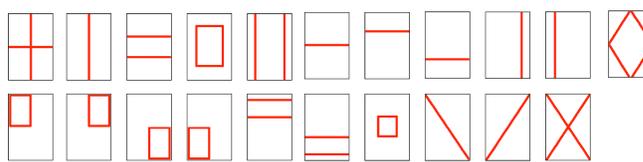


Figure 14. Editorial Design Templates used in Experiment-2.

B. Experiment-1: Experimental Results

In this section, we evaluate the retrieval precision of the five queries specified in the previous subsection. Figure 11 shows some retrieval results for a combination of a query and color scheme. Our approach assigns two scores for the retrieved rank. The first is the average number of correct answers in the top 20 that were retrieved using only template relevance. The second score is the average rank of correct answer data that has the chosen color scheme. Our approach assigns a score based on the retrieved ranking. Thus, we obtain one score for the average correct ranking using layout relevance alone (i.e., from search method 1) and one for the average rank of correct answer data when color scheme templates were integrated (i.e., search method 2).

Figure 12 shows the average scores of the top 20 ranking for each query. This result shows that search method 2 (integrating both layout and color templates) achieves better retrieval precision than method 1 (using the layout template only), as a superior (lower) rank has been assigned to results that correspond to the query. This shows that our system effectively retrieves e-book images using layout templates and color schemes.

C. Experiment-2: Outline of Experimental Studies

In this section, we evaluate the effectiveness of our system by measuring the time consumed for retrieving the desired book by using our system and the conventional book search system. The aim of this experiment is to clarify the effectiveness of our visual query expansion method by comparing it with the exiting book retrieval system. We have compared our system and the keyword-based search engine implemented in Amazon.co.jp. It is important to mention that this experiment assumes that a user forgets the details about the target book and cannot submit the book's title correctly. In such a situation, our visual query expansion method

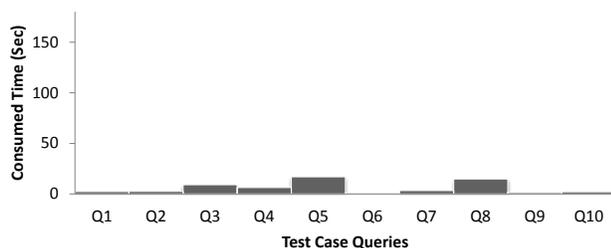


Figure 15. Time Consumed to Retrieve the Correct Result by using Query-by-Appearance (Method-1).

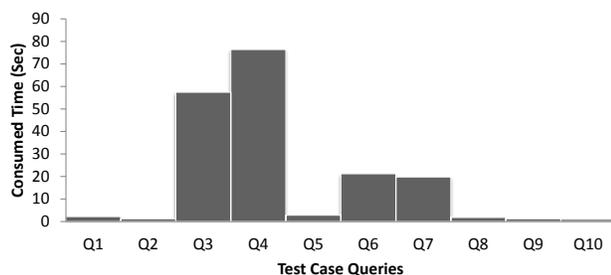


Figure 16. Time Consumed to Retrieve the Correct Result by Submitting an Author's Name as a Query for Amazon.co.jp (Method-2).

effectively contributes to user's retrieval process. We compare the time consumed to find the correct result by using four search methods: 1) queries are generated using Query-by-Appearance system, 2) publisher's name is submitted for Amazon.co.jp, 3) author's name is submitted for Amazon.co.jp, and 4) publication date is submitted for Amazon.co.jp. We have asked five casual people for five test subjects to find ten books by using those four methods. Images of ten books are shown in Figure 13. Figure 14 shows the editorial design templates used in Experiment-2. We have measured the time (second) until each test subject find the correct result. We show that our visual query expansion method and visualization method make a significant contribution to reduce the time required to find out the correct result.

D. Experiment-2: Experimental Results

In this section, we evaluate the retrieval time of the ten queries by using four methods specified in the previous subsection. We plot the averaged time of five test subjects for each query. When a test subject could not find the correct result, or gave up finding the book, we considered it as consumption of 180 seconds. Figure 15 shows the time consumed to find the correct result by using Query-by-Appearance system. Figure 16 shows the time consumed to find the correct result by submitting author's name into Amazon.co.jp. Figure 17 shows the time consumed to find the correct result by submitting publisher's name into Amazon.co.jp. Figure 18 shows the time consumed to find the correct result by submitting publication date into Amazon.co.jp.

The average time consumed for searches carried out using our Query-by-Appearance system (Method-1) was 6

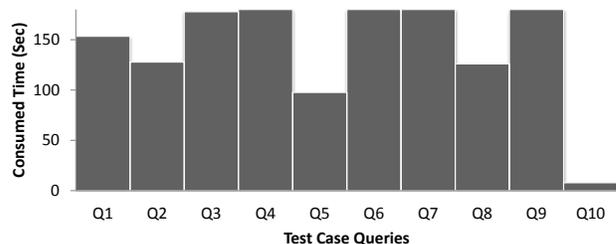


Figure 17. Time Consumed to Retrieve the Correct Result by Submitting a Publisher's Name as a Query for Amazon.co.jp (Method-3).

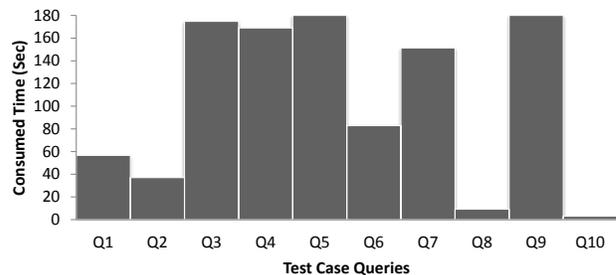


Figure 18. Time Consumed to Retrieve the Correct Result by Submitting a Publication Date as a Query for Amazon.co.jp (Method-4).

seconds; the average time consumed for searches carried out using author's name (Method-2) was 19 seconds; the average time consumed for searches carried out using publisher's name (Method-3) was 104 seconds; the average time consumed for searches carried out using publication date (Method-4) was 141 seconds. This confirms that our visual query expansion method and visualization method significantly improve the e-book retrieval efficiency when it is compared with the existing search engine.

IX. CONCLUSIONS AND FUTURE WORK

This paper proposed the *Query-by-Appearance* system for e-books. The system provides an intuitive and effective query input and visual retrieval method based on the similarity of the overall layout and color scheme. A unique feature of this system is its visual query expansion function, which exploits structural design and analysis from the field of editorial design. This assistant mechanism is intuitive because it allows users to find a desired e-book by submitting a simple query. Our visual query expansion function is a design-based template-matching function that compares a user's rough query with the embedded editorial design templates. We have performed an evaluation of the effects of our visual query expansion method on realistic search tasks. Experimental results showed that visually expanded queries perform a better and more consistent performance than manually conducted queries. In future work, we plan to develop a prototype system that supports the full specifications of EPUB3, and to perform a feasibility study by evaluating the scalability and effectiveness of our approach when applied to existing e-books.

ACKNOWLEDGMENT

This research was supported by SCOPE: Strategic Information and Communications R&D Promotion Programme of the Ministry of Internal Affairs and Communications, Japan: “Kansei Time-series Media Hub Mechanism for Impression Analysis / Visualization / Delivery Intended for Video / Audio Media.”

REFERENCES

- [1] Koike, Y. and Kurabayashi, S., “Query-by-Appearance System for Style-Oriented Media Retrieval,” In Proceedings of the Seventh International Conference on Internet and Web Applications and Services (ICIW 2012), pp.290–295, Stuttgart, Germany, May 27-June 1, 2012.
- [2] Gartner, “Gartner Says Worldwide PC Shipments in First Quarter of 2011 Suffer First Year-Over-Year Decline in Six Quarters,” 13 April 2011, <http://www.gartner.com/it/page.jsp?id=1632414>.
- [3] IDC, “Nearly 18 Million Media Tablets Shipped in 2010 with Apple Capturing 83% Share; eReader Shipments Quadrupled to More Than 12 Million, According to IDC,” 10 March 2011, <http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22737611>.
- [4] American Association of Publishers, “Publishing Industry Has Strong January Revenue Growth In Print Books And Ebooks For All Audiences,” 2012, <http://www.publishers.org/press/62/>, (accessed 2012-8-6).
- [5] Datta, R., Joshi, D., Li, J., Wang, J.Z., “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Comput. Surv.* 40, pp. 1–60, 2008.
- [6] Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R., “Content-based image retrieval at the end of the early years,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, pp.1349–1380, 2000.
- [7] Kherfi, M.L., Ziou, D., Bernardi, A., “Image Retrieval from the World Wide Web: Issues, Techniques, and Systems,” *ACM Computing Surveys*, 36, pp. 35–67, 2004.
- [8] Shipman, F.M., Marshall, C.C., Moran, T.P., “Finding and using implicit structure in human-organized spatial layouts of information,” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI’95. pp. 346–353. ACM, 1995.
- [9] Woodruff, A., Faulring, A., Rosenholtz, R., Morrisson, J., Pirolli, P., “Using thumbnails to search the Web,” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI’01. pp. 198–205. ACM, 2001.
- [10] Abdullah, N., Gibb, F., “Using a Task-Based Approach in Evaluating the Usability of BoBIs in an E-book Environment,” In Proceedings of the 30th European Conference on Information Retrieval, pp. 246–257. Springer-Verlag, 2008.
- [11] Kudělka, M., Takama, Y., Snášel, V., Klos, K., Pokorný, J., “Visual Similarity of Web Pages,” In Proceedings of the 6th Atlantic Web Intelligence Conference - AWIC 2009, pp. 135–146. Springer-Verlag, 2010.
- [12] Jing, Y. and Baluja, S., “PageRank for Product Image Search,” In Proceedings of the 17th International Conference on World Wide Web, pp. 307-316, 2008.
- [13] Tang, Y.Y., Yan, C.D., and Suen, C.Y., “Document Processing for Automatic Knowledge Acquisition,” *IEEE Trans. on Knowledge and Data Engineering*, 6(1), pp. 3–21, 1994.
- [14] Gao, L., Tang, Z., Lin, X., Liu, Y., Qiu, R., and Wang, Y., “Structure extraction from PDF-based book documents,” In Proceeding of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries - JCDL’11, pp. 11–20, 2011.
- [15] He, F., Ding, X., and Peng, L., “Hierarchical Logical Structure Extraction of Book Documents by Analyzing Tables of Contents,” In Proc. of the International Conference on Document Recognition and Retrieval XI, pp. 6–13, 2004.
- [16] Lin, C., Niwa, Y., and Narita, S., “Logical Structure Analysis of Book Document Images Using Contents Information,” In Proc. of the 4th International Conference on Document Analysis and Recognition, ICDAR’97, pp.1048–1054, 1997.
- [17] Igarashi, T., Matsuoka, S., Masui, T., “Adaptive recognition of implicit structures in human-organized layouts,” *Proceedings of the 11th IEEE International Symposium on Visual Languages*, pp. 258–266, 1995.
- [18] Lee, K.H., Choy, Y.C. and Cho, S.B., “Logical Structure Analysis and Generation for Structured Documents: A Syntactic Approach,” *IEEE Transaction on Knowledge and Data Engineering*, 15(5), pp.1277–1294, 2003.
- [19] Harman, D., “Relevance feedback revisited,” In Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1–10, 1992.
- [20] Koolen, M., Kazai, G., Craswell, N., “Wikipedia pages as entry points for book search,” In Proceedings of the Second ACM International Conference on Web Search and Data Mining - WSDM ’09. pp. 44-53. ACM Press, 2009.
- [21] Fonseca, B.M., Golgher, P., Póssas, B., Ribeiro-Neto, B., Ziviani, N., “Concept-based interactive query expansion,” In Proceedings of the 14th ACM International Conference on Information and Knowledge Management - CIKM’05, pp. 696, 2005.
- [22] Hoque, E., Strong, G., Hoerber, O., Gong, M., “Conceptual Query Expansion and Visual Search Results Exploration for Web Image Retrieval,” In Proceedings of the 7th Atlantic Web Intelligence Conference, AWIC 2011, pp. 73–82. Springer-Verlag, 2011.
- [23] Natsev, A. (Paul), Haubold, A., Tešić, J., Xie, L., Yan, R., “Semantic concept-based query expansion and re-ranking for multimedia retrieval,” In Proceedings of the 15th ACM International Conference on Multimedia, pp. 991–1000, 2007.
- [24] Joly, A., Buisson, O., “Logo retrieval with a contrario visual query expansion,” In Proceedings of the 17th ACM International Conference on Multimedia, pp. 581–584, 2009.
- [25] Totogawa, Y., “Color Scheme & Color Design,” ISBN: 978-4797359237, Softbank Creative, 2012 (In Japanese).
- [26] Newhall, S.M., Nickerson, D., and Judd, D.B., “Final Report of the O.S.A. Subcommittee on the Spacing of the Munsell Colors,” *Journal of the Optical Society of America*, 33(7), pp. 385-411, 1943.
- [27] Godlove, I.H. “Improved Color-Difference Formula, with Applications to the Perceptibility and Acceptability of Fadings,” *Journal of the Optical Society of America*, Vol. 41, NO. 11, pp.760-770, 1951.
- [28] Luo, M.R., Cui, G., and Rigg, B., “The development of the CIE 2000 colour-difference formula: CIEDE2000,” *Color Research & Application*, 26(5), pp. 340-350, 2001.

Performance Test Case Generation for Java and WSDL-based Web Services from MARTE

Antonio García-Domínguez and Inmaculada Medina-Bulo
Department of Computer Languages and Systems
University of Cádiz
Cádiz, Spain
{antonio.garciadominguez, inmaculada.medina}@uca.es

Mariano Marcos-Bárcena
Department of Industrial Design and Mechanical Engineering
University of Cádiz
Cádiz, Spain
mariano.marcos@uca.es

Abstract—Obtaining the expected performance from a workflow would be easier if every task included its own specifications. However, normally only global performance requirements are provided, forcing designers to infer individual requirements by hand. Previous work presented two algorithms that automatically inferred local performance constraints in Unified Modelling Language activity diagrams annotated with the Modelling and Analysis of Real-Time and Embedded Systems profile. This work presents an approach to use these annotations to generate performance test cases for multiple technologies, linking a performance model and a design model with a weaving model in a meet-in-the-middle approach so users can write their software according to their needs. Two implementations of the approach are described, which have been published as open source software. The first implementation extracts models from Java unit tests using MoDisco and weaves them with the performance models in order to convert some or all of the test cases in the selected JUnit test suites to performance tests. The second implementation extracts models from WSDL documents including message templates and template variables and uses these models for generating test inputs, test code and test infrastructure. Users can customise the service catalogues, test inputs and message templates to obtain a high degree of flexibility, and the test infrastructure provides automated configuration and test reports. These two successful implementations for different technologies validate the proposed approach as a generic framework for generating performance test case artefacts from existing software.

Keywords—software performance; Web Services; MARTE; model driven engineering; test generation.

I. INTRODUCTION

Software needs to meet both functional and non-functional requirements. Performance requirements are among the most commonly used non-functional requirements, and in some contexts they can be just as important as functional requirements. In addition to soft and hard real-time systems, Service Oriented Architectures (SOAs) must be considered as well. Within SOAs, it is common practice to sign Service Level Agreement (SLAs) with external services, to compensate consumers in case of problems. It is also quite common to create “service compositions”, which are services that integrate several lower level services (normally, Web Services from external providers). However, it may be difficult to establish what performance level should be required from the composed services. Too little, and the performance requirements for the composition will not be met. Too much, and the provider may

charge more than desired. In addition, developers must test the external services to ensure that they can provide the required performance levels.

There is a large variety of proposals for estimating the required level of performance and measuring the actual performance of a system. Measurements can be used for detecting performance degradations over time, identifying load patterns or checking the SLAs. However, the requirements set by the SLA are usually broad and cover a large amount of functionality: when violated, it might be hard to pinpoint the original cause. Whenever possible, performance requirements should be as specific as possible, but that would be too expensive for all but the most trivial systems.

This paper is an extended version of our previous work [1]. This previous work presented an overall approach for using the models produced by the inference algorithms in [2] to generate performance artefacts for multiple target technologies. The algorithms can “fill in the blanks” for the response time and throughput requirements of every activity in the model, starting from a global annotation and some optional local annotations set by the user. Originally, users would then have to write the actual performance tests manually, taking the results produced by these algorithms as a reference. However, writing these tests for every part of a reasonably-sized system could incur in a considerable cost: ideally, it should be partly automated.

The present work provides an up-to-date account of the results obtained after implementing the two approaches outlined in our previous work. Though the overall approach has not changed much, many details had to be revised as the tools were defined. For instance, the Java approach now offers finer-grained control on the tests to be used and can use other metrics apart from maximum response times, such as averages, medians or percentiles. It is the WSDL-based approach that has changed the most: WSDL documents have been found to be too complex for doing model weaving directly on them. Instead, a custom model extractor has been developed, which is combined with a test generator and a template language to generate the input messages.

The rest of this work is structured as follows. After discussing related work in Section II, the MARTE profile is introduced in Section III and the performance models are presented in Section IV. Section V describes the general

approach for generating test artefacts. This approach is applied to reusing Java unit tests as performance tests using the ContiPerf library in Section VI. Section VII is dedicated to generating performance tests (input data, testing code and infrastructure) from any Web Service described using WSDL. Finally, conclusions and future lines of work are listed in Section VIII.

II. RELATED WORK

According to Woodside et al. [3], performance engineering comprises all the activities required to meet performance requirements. These activities include defining the requirements, analysing early performability models (such as layered queuing networks [4] or process algebra specifications [5]) or testing the performance of the actual system. Our previous work in [2] focused on helping the user define the requirements using MARTE-annotated [6] UML activity diagrams as notation. The present work will show how to assist the user in creating the performance test artefacts from the resulting requirements.

Many testing approaches do not work directly with the implemented system, but rather with a simplified representation (a model). There is a large number of works dealing with model-based testing, i.e., “the automatable derivation of concrete test cases from abstract formal models, and their execution” [7]. Most of them (as evidenced by [7] itself) are dedicated to functional testing: the rest of this section will focus on those dedicated to model-based performance testing.

Barna et al. present in [8] a hybrid approach, which uses a 2-layered queuing network (LQN) to derive an initial stress workload for a website. This workload is used to test the system and refine the original LQN model in a feedback loop that searches for the minimum load that would make the system violate one of its performance constraints. Like our work, it combines the analysis of a model with the execution of a set of test cases. However, its goal is completely different: the algorithms in [2] intend to define the appropriate quality service levels for the individual services in order to meet the desired quality service level of the entire workflow, whereas this approach would estimate the maximum workload that a workflow could handle within a certain quality service level.

Di Penta et al. show in [9] another approach with the same goal of finding workloads that induce service level agreement violations. However, they use genetic algorithms instead of a LQN model and test WSDL-based Web Services instead of a regular website.

Suzuki et al. have developed a model-based approach for generating testbeds for Web Services [10]. SLA and behaviour models are used to generate stubs for the external services used by the service. This allows users to check that their own services can work correctly and with the expected level of performance as long as the external services meet their SLAs. However, this approach does not generate input messages for the services themselves. Still, we could use this work to check the validity of the performance constraints inferred by the algorithms in [2] in combination with the approach which

will be presented in Section VII, by replacing all services in the workflow with stubs and testing the performance of the composition.

As illustrated by the above references, there is a wealth of methods for generating performance test cases and testbeds for Web Services. However, we have been unable to find another usage of model weaving for generating performance test artefacts for multiple technologies. This is in spite of the fact that model composition using model weaving has been used regularly ever since the authors of the original ATLAS Model Weaver proposed it [11]. For instance, Vara et al. use model composition to decorate their extended use case models with additional information required for a later transformation [12].

III. THE MARTE PROFILE

UML is widely used as a general purpose modelling language for software systems. However, UML cannot model non-functional aspects such as performance requirements.

For this reason, the OMG (Object Management Group) proposed in 2005 the SPT (Schedulability, Performability and Time) profile [13], which extended UML with a set of stereotypes describing scenarios that various analysis techniques could take as inputs. In 2008, OMG proposed the QoS/FT (Quality of Service and Fault Tolerance Characteristics and Mechanisms) profile [14], with a broader scope than SPT and a more flexible approach: users formally defined their own quality of service vocabularies to annotate their models.

When UML 2.0 was published, OMG saw the need to update the SPT profile and harmonise it with other new concepts. This resulted in the MARTE (Modelling and Analysis of Real-Time and Embedded Systems) profile [6], published in 2009. Like the QoS/FT profile, the MARTE profile defines a general framework for describing quality of service aspects. The MARTE profile uses this framework to define a set of pre-made UML stereotypes, as those in the SPT profile.

The rest of this section presents the architecture of the MARTE specification and focuses on the key subset that has been used for the performance models.

A. Architecture

The MARTE profile is a complex specification, spanning over 700 pages. It is organised into several subprofiles and includes a normative model library with predefined types and concepts and an embedded expression language known as the Value Specification Language (VSL). Figure 1 lists each of the packages that constitute MARTE and their elements:

- The “MARTE foundations” package defines the core concepts that are used throughout the other profiles, such as the concept of a non-functional property (NFP) or how to model time, resources (using the General Resource Modelling or GRM subprofile) or the allocation of functional elements on the available resources.
- The “MARTE analysis model” package is used to annotate application models to support analysis of system properties. The Generic Quantitative Analysis Modelling

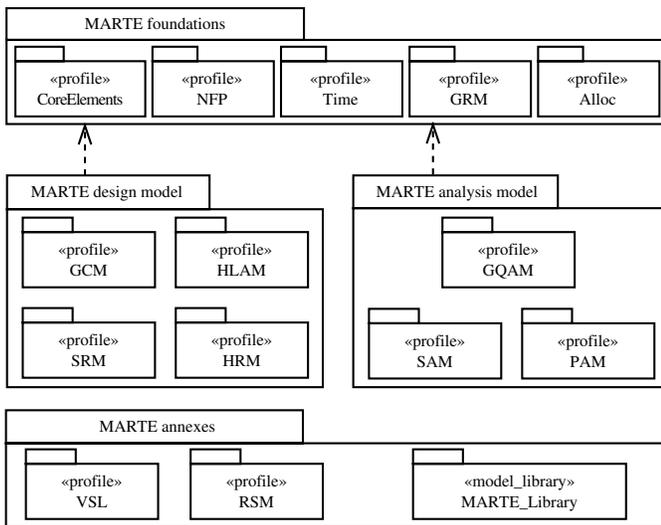


Fig. 1. Architecture of the MARTE profile [6]

(GQAM) subprofile uses the foundations package and the normative model library to provide a base set of concepts for the two kinds of analysis supported by MARTE: schedulability analysis and performance analysis. Schedulability analysis predicts whether a set of software tasks meets its timing constraints and is modelled using the Schedulability Analysis Modelling (SAM) subprofile. Performance analysis determines whether a system with non-deterministic behaviour can provide adequate performance, and is supported through the Performance Analysis Modelling (PAM) subprofile.

- The “MARTE design model” package provides the required concepts for modelling the features of real-time and embedded (RT/E) systems. The Generic Component Modelling (GCM) subprofile provides additional core concepts for RT/E systems. The High-Level Application Modelling (HLAM) subprofile provides the concept of a real-time execution unit that manages several resources and a queue of messages with various real-time requirements. Finally, the Detailed Resource Modelling (DRM) subprofile provides facilities for describing the software and hardware resources used by the system.
- The “MARTE annexes” package includes the Value Specification Language (VSL) used for all MARTE expressions, the Repetitive Structured Modelling (RSM) package for describing available software and hardware parallelism, and the normative MARTE model library. The normative MARTE library defines the set of standard primitive types (such as real numbers or integers) and derived types (such as vectors of integers or NFPs involving a real value), among many other concepts beyond the scope of this article.

B. GQAM

Using the Generic Quantitative Analysis Modelling (GQAM) subprofile requires the definition of an *AnalysisContext*,

which is formed by a *WorkloadBehavior* object (the workload to be run) and a *ResourcesPlatform* object (the resources to be used). An *AnalysisContext* may also include a set of user-defined context parameters, which will be available as variables in the VSL expressions of the NFP.

The workload is then divided into the *WorkloadEvent* describing the request arrival pattern, and the *BehaviorScenario* specifying how these requests should be handled and the NFPs for them. A *BehaviorScenario* is further divided into *Steps* which are ordered using *PrecedenceRelations* of several kinds, such as sequential, branching, merging, forking or joining relations. Each *Step* may have NFPs of its own. These NFPs include response time, throughput, utilisation or the expected number of repetitions.

Finally, the GQAM concepts are mapped to UML stereotypes. For instance, *AnalysisContext*, *BehaviourScenario* and *Step* are mapped to the `«GaAnalysisContext»`, `«GaScenario»` and `«GaStep»` stereotypes, respectively.

C. VSL

As mentioned above, the GQAM *BehaviorScenario* and *Step* classes can contain NFPs for many aspects. However, properly describing the value of a NFP requires more than a simple scalar value: it is required to describe aspects such as measurement sources, measurement unit, precision and so on. In addition, the value of a NFP may be derived from a complex expression using several context parameters. All these features can be described using the Value Specification Language (VSL) embedded within the MARTE profile.

VSL provides a set of datatypes that extends the primitive types available in UML with composite types (such as intervals, collections or tuples) and subtypes. It also provides a textual syntax for complex expressions that may use conditional operators, invoke operators, compute time values and use arithmetic operators, among other features. Both can be combined: for instance, $(expr=2+3*f, ms, req)$ is a VSL tuple that represents a duration in milliseconds (*ms*) that has been required by the developer (*req*) and is computed from the *f* context parameter as $2 + 3f$.

IV. PERFORMANCE MODELS

This section will present the notation used by the performance algorithms described in [2]. The models are used for performance analysis, and so PAM would appear to be the best starting point. However, the focus of the algorithms is different than the one favoured by PAM, which is predicting the performance of the whole system from its parts. Instead, the algorithms infer the performance needed in each part of the system from the global requirements. For this reason, it only uses the generic analysis core, the GQAM subprofile.

To keep the models simple, the notation only uses the three stereotypes in Section III-B. Due to the additional complexity in explicitly describing the precedence relations among the *Steps*, this information is inferred from the flows in the UML models.

Figure 2 shows a simple example. Inferred annotations are highlighted in bold:

- 1) The activity is annotated with a `<<GaScenario>>` stereotype, in which `respT` specifies that every request is completed within 1 second, and `throughput` specifies that 1 request per second needs to be handled. These expressions have their `source` attribute set to `req`, as they represent explicit requirements from the developer.
- 2) In addition, the activity declares a set of context parameters in the `contextParam` field of the `<<GaAnalysisContext>>` stereotype. These variables represent the time per unit of weight that must be allocated to their corresponding activity in addition to the minimum required time. Their values are computed by the time limit inference algorithm.
- 3) Each action in the activity is annotated with `<<GaStep>>`, using in `hostDemand` a VSL expression of the form $m + ws$, where m is the minimum time limit, w is the weight of the action for distributing the remaining time, and s is the context parameter linked to that action. These expressions also have their `source` attribute set to `req`, for the same reasons as those in `<<GaScenario>>`.

The time limit inference algorithm adds a new constraint to `hostDemand`, indicating the exact time limit to be enforced. The throughput inference algorithm extends `throughput` with a constraint that lists how many requests per second should be handled. As these constraints have been automatically inferred, their `source` attribute is set to `calc` (calculated).

- 4) Outgoing edges from condition nodes also use `<<GaStep>>` but only for the `prob` attribute, which is set by the user to the estimated probability it is traversed.

V. OVERALL APPROACH

The model shown in the previous section is entirely abstract: at that level of detail, it cannot be executed automatically. It will have to be implemented through other means.

After it has been implemented, it would be useful to take advantage of the original model to generate the performance test cases. However, the model lacks the required design and implementation details to produce executable artefacts. To solve this issue, several approaches could be considered:

- 1) The abstract model could be extended with additional information, but that would clutter it and make it harder to understand.
- 2) On the other hand, the implementation models could be annotated with performance requirements, but this would also pollute their original intent.
- 3) Finally, a separate model that links the abstract and concrete models could be used. This is commonly known as a *weaving model*. Several technologies already exist for implementing these, such as AMW [11] or Epsilon ModelLink [15]. While AMW uses a generic weaving metamodel, ModelLink is a more lightweight approach

that requires defining custom weaving metamodels for every pair of metamodels.

In order to preserve the cohesiveness of the abstract performance model and the design and implementation models, the third approach has been chosen. The weaving model will need to allow users to annotate the links with the additional information required by the testing process, the target technologies and the generation process itself. With target technologies, we refer not only to the performance testing framework or tool which will run the generated tests, but also all the components which will be part of the test infrastructure. As we will see in Section VII, this may include IDEs (e.g. Eclipse) or build automation tools (Maven).

Some of the information may be shared by a set of tests (possibly all of them), and some of the information will be specific to a particular link between a design/implementation artefact and a performance requirement. For instance, while the number of threads used to exercise the system under test may need to be the same for all the tests, the interpretation of the time limit requirement as a median, an average or a percentile may change from test to test.

After establishing the required links, the next step is generating the tests themselves. To do so, a regular Model-to-Text (M2T) transformation could be used, written in a specialised language such as the Epsilon Generation Language [16]. In case it were necessary to slightly refine or validate the weaving model before, an intermediate Model-to-Model (M2M) transformation could be added. Figure 3 illustrates the models and steps involved in the overall approach.

In some cases, we may want to allow users to easily customise certain interesting parts of the tests, while abstracting them from the details that are less interesting. These interesting parts could be written into a custom domain-specific language instead of code, which would be interpreted as the tests were executed by augmenting the testing infrastructure accordingly. We will see an example of this with the TestSpec language later in Section VII-E.

The next sections will show two applications of the overall approach in Figure 3, using different technologies to assist in generating performance test artefacts in different environments. Both approaches have been implemented and are freely available under the open source Eclipse Public License at [17]. In order to develop these transformations, a bottom-up approach was used: a manually developed performance test environment was gradually replaced by automatically generated fragments until only the weaving model remained. After the entire process had been automated, the generators were refined to allow for more flexibility and convenience.

VI. REUSING JAVA UNIT TESTS AS PERFORMANCE TESTS

Generating executable performance test cases from scratch automatically will usually require many detailed models and complex transformations, which are expensive to produce and maintain. The initial effort required may deter potential adopters. An alternative inexpensive approach is to repurpose existing functional tests as performance tests as a starting

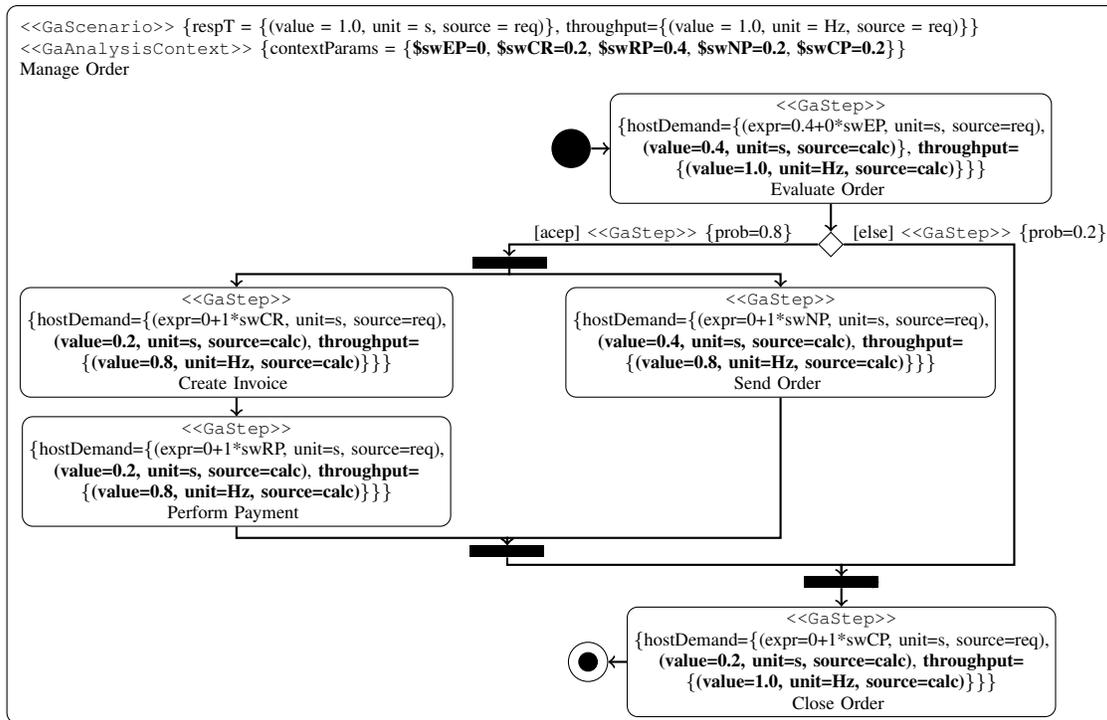


Fig. 2. Simple example model annotated by the performance inference algorithms

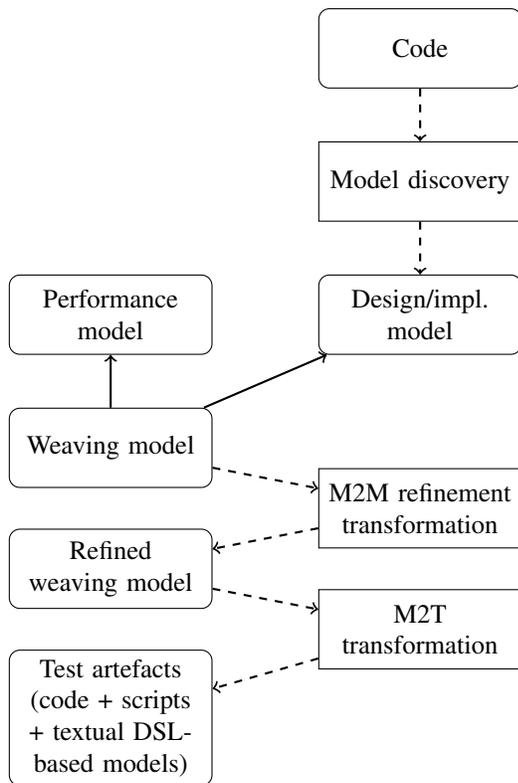


Fig. 3. Overall approach for generating performance test artefacts from abstract performance models

```

@RunWith(ContiPerfSuiteRunner.class)
@SuiteClasses(TFunctionalJUnit4.class)
@PerfTest(invocations = 100, threads = 10)
@Required(max=1000)
public class InferredLoadTest {}
    
```

Listing 1. Java code for wrapping the TFunctionalJUnit4 JUnit 4 test suite using ContiPerf

point. This is the aim of libraries such as ContiPerf [18]. The rest of the section will show the overall approach in Figure 3 was customised for this particular use case. The resulting transformation chain is shown in Figure 4.

A. Target framework: ContiPerf

Listing 1 shows how ContiPerf is normally used. Instead of using Java objects, ContiPerf uses Java 6 annotations, which are easier to generate automatically. The @PerfTest annotation indicates that the test will be run 100 times using 10 threads, so each thread will perform 10 invocations. @Required indicates that each of these invocations should finish within 1000 milliseconds at most. @SuiteClasses points to the JUnit 4 test suites to be reused for performance testing, and @RunWith tells JUnit 4 to use the ContiPerf test runner.

B. Model extraction

In both cases, the code itself is straightforward to generate. However, the generated code must integrate correctly with the existing code. If the code was not produced using a model-driven approach, there will not be a design or implementation

model to link to. Instead, a model of the structure of the existing code is derived using the Eclipse MoDisco model discovery tool [19]. Eclipse MoDisco can generate models from Java code such as that shown in Figure 5.

C. Weaving metamodel

Once the performance and the implementation models have been produced, the next step is to link them using a new *weaving model* that conforms to the metamodel in Figure 6. Some of the types in the weaving metamodel refer to types in the *uml* and *java* packages from the UML2 metamodel and the MoDisco Java metamodel, respectively.

Each model consists of an instance of *PerformanceRequirementLinks*, which provides several global configuration parameters and contains a set of *PerformanceRequirementLink* instances. Users can set the number of samples which should be collected for each test, the number of threads over which these should be distributed and the directory under which the code should be generated. Every link relates an UML *ExecutableNode* with a Java class: if no *MethodDeclarations* are specified, all tests will be reused. Otherwise, only the selected methods will be reused. Finally, the target time limit may be enforced as a maximum value (MAX), average (AVERAGE), median (MEDIAN) or a percentile (the rest).

Originally, the models referenced the MARTE «GaStep» stereotype instead of the UML *ExecutableNode*. These references were switched to *ExecutableNode* as the «GaStep» stereotype was optional if the default minimum time limit $m = 0$ and weight $w = 1$ were used.

D. Code generation

Models are populated by combining the standard Epsilon Modeling Framework (EMF) tree-based editors and the three-

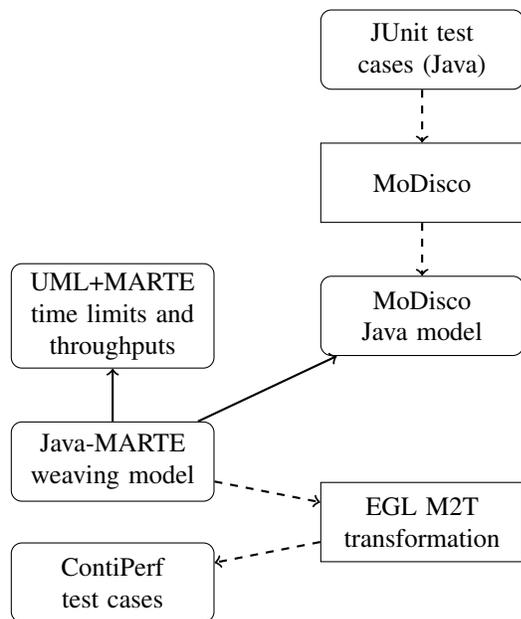


Fig. 4. Instance of the overall approach for wrapping JUnit tests into ContiPerf tests

```

@Required(throughput=2, max=400)
public class WrapSomeTests extends OriginalSuite {
    @Rule public MethodRule f =
        new FilterByClassRule(this.getClass());

    @Rule public ContiPerfRule i = new ContiPerfRule();

    @PerfTest(invocations=1000, threads=5)
    @Test @Override
    public void first() throws Exception {
        super.first();
    }

    // protected region customTests off begin
    // Add your own tests here
    // protected region customTests end
}
  
```

Listing 2. Java code wrapping one test from *OriginalSuite* using *ContiPerf*

```

@WebService
public class HelloWorld {
    @WebMethod
    public String greet(
        @WebParam(name="name") String name)
    {
        return "Hello_" + name;
    }
}
  
```

Listing 3. Java code using JAX-WS for a "HelloWorld" Web Service

pane Epsilon ModelLink editor (as in Figure 7). ModelLink provides a drag-and-drop approach to model linking that is convenient for model weaving. The EMF editors have been manually customised so users may only pick JUnit 4 test suites and test methods.

The code is generated using a set of Epsilon Generation Language (EGL) templates. When all tests are reused as performance tests, the generated code will use the *ContiPerfSuiteRunner* test runner, as in Listing 1.

However, when only some tests are wrapped the code will resemble that in Listing 2. The *ContiPerfRule* would normally convert all tests into performance tests. By using the *FilterByClassRule* helper class (also generated with EGL), the generated code will be able to specify that only some of those tests need to be reused as performance tests.

VII. GENERATING PERFORMANCE TESTS FOR WSDL-BASED WEB SERVICES

In the previous section, the approach was applied to existing JUnit test cases, repurposing them as performance test cases. This section will discuss how to generate performance test artefacts for a Web Service (WS) [20] in a language agnostic manner. The implemented solution is summarised in Figure 8.

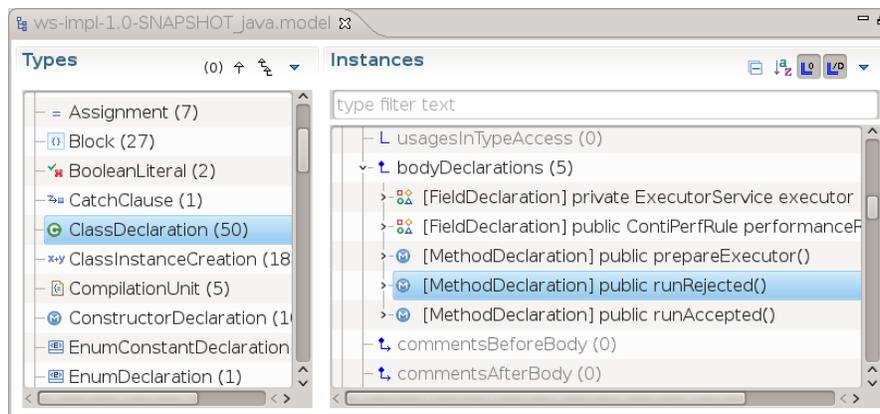


Fig. 5. MoDisco model browser showing a model generated from an Eclipse Java project

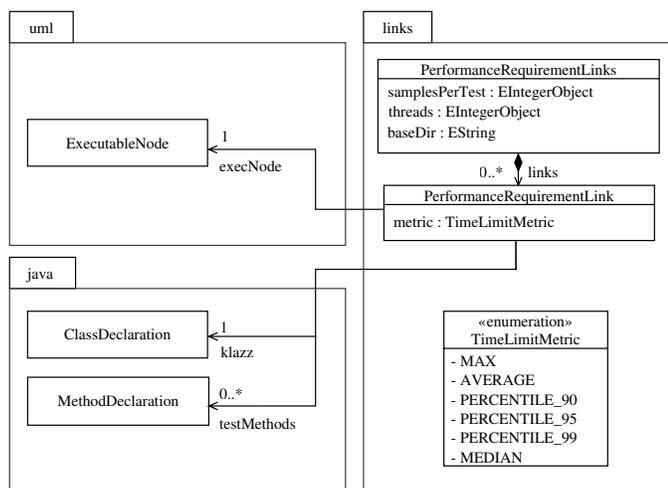


Fig. 6. Java-MARTE weaving metamodel

A. Motivation

Web Services based on the WS-* technology stack are usually described using a Web Services Description Language (WSDL) [21] document. This XML-based document is an abstract and language-independent description of the available operations for the service and the messages to be exchanged between the service and its consumers.

Existing Web Service frameworks such as Apache CXF [22] can generate most of the code required to implement and consume the services from the WSDL document. Users only need to implement the business logic of the services. In addition, some frameworks (CXF included) can work in reverse, generating WSDL from adequately annotated code.

Listing 3 shows an example fragment of Java code that implements a simple “HelloWorld” Web service using standard JAX-WS [23] annotations. This Java code could be tested using the approach in Section VI. However, a WSDL-based approach would be easier to work with when mixing services written in multiple languages or frameworks.

B. Target performance testing tool: The Grinder

The previous section reused unit tests written in a particular language (Java) and a particular framework (JUnit). Therefore, the target technology was an extension upon this framework (ContiPerf). However, since the WSDL description of a Web Service does not depend on the language that it is implemented in, we are not limited to a specific language for the tests. Instead, we will use a dedicated performance testing tool. Such tools help define tests with less cost and in a way that is independent of the implementation language of the software under test.

We evaluated the following tools based on the ease with which test specifications could be generated for them, by developing a simple performance test on a single service with each of them and studying the files required by the tools:

- The Grinder [24] used textual configuration files to configure the test environment, which executes Jython scripts that use the public API provided by the tool.
- Apache JMeter [25] used reflective XML documents. Most of their contents were directly translated into API calls of the underlying Java code, tightly coupling the transformation to their internal code structure.
- Eviware loadUI [26] had the most complicated input format out of the three. It used both binary and textual artefacts. Some of the textual artefacts were trees of Java classes, which would have to be generated and then packed together with the binary parts.

The Grinder [24] was selected among the available tools, as its input format was the easiest to generate and provided more flexibility.

In addition, The Grinder is easy to scale up depending on the testing requirements. The Grinder can launch several processes that spawn a certain number of threads which will repeatedly run the test. It can also optionally distribute work over several machines: one of them provides a graphical console and acts as the master, and the rest are agents that manage a set of worker processes.

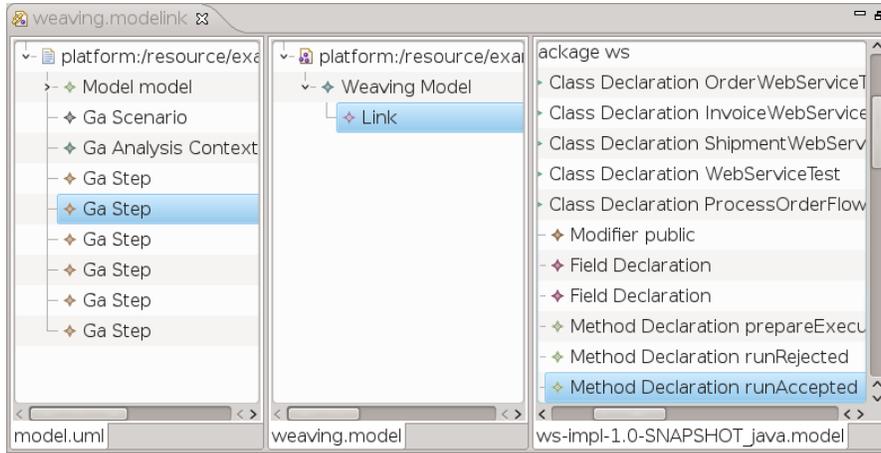


Fig. 7. Screenshot of the Epsilon ModelLink editor weaving the MARTE performance model and the MoDisco model

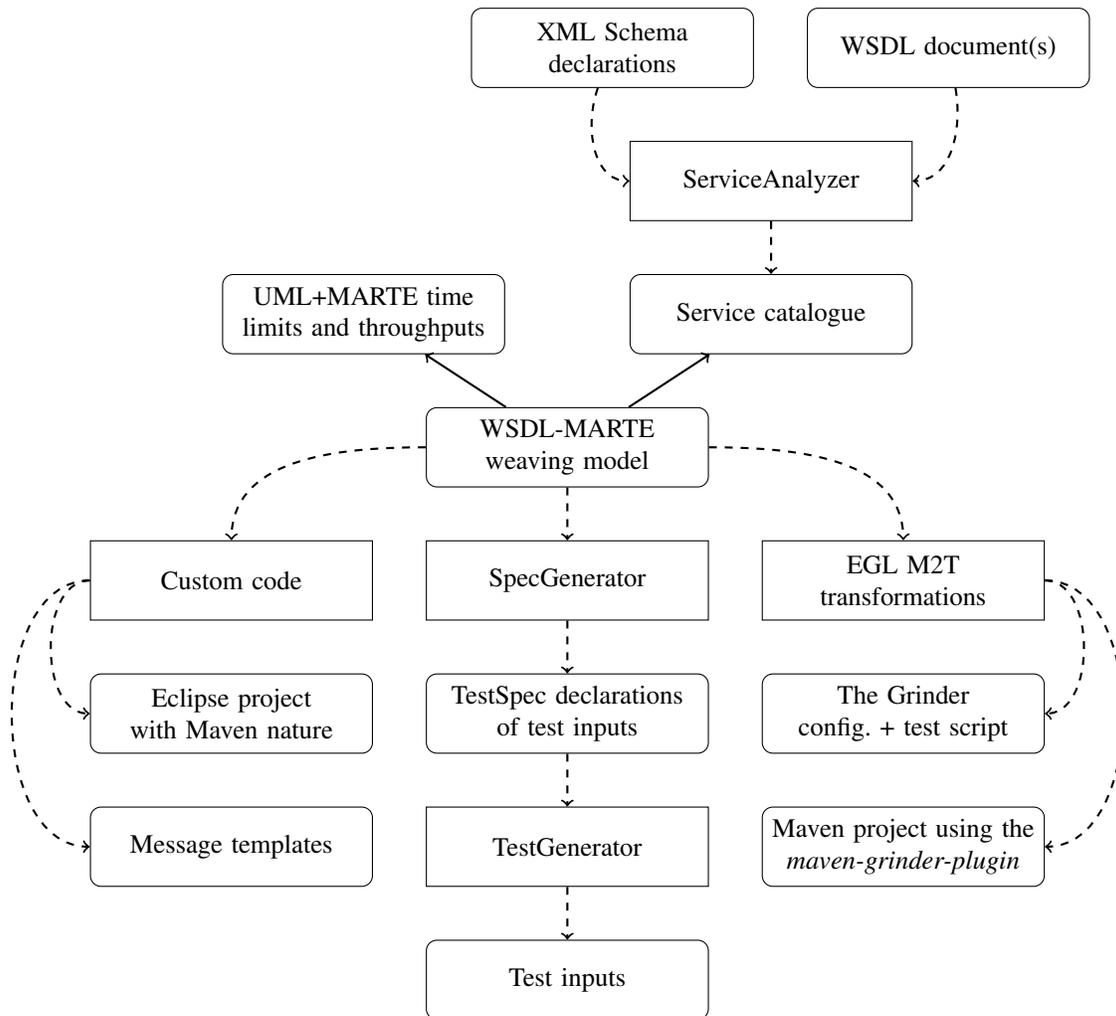


Fig. 8. Instance of the overall approach for generating performance tests for WSDL-based Web Services. In comparison with the approach specifically targeted for Java, this approach requires integrating several technologies, such as a build automation tool (Maven), three custom tools (ServiceAnalyzer, SpecGenerator and TestGenerator) and a dedicated performance testing tool (The Grinder), among others.

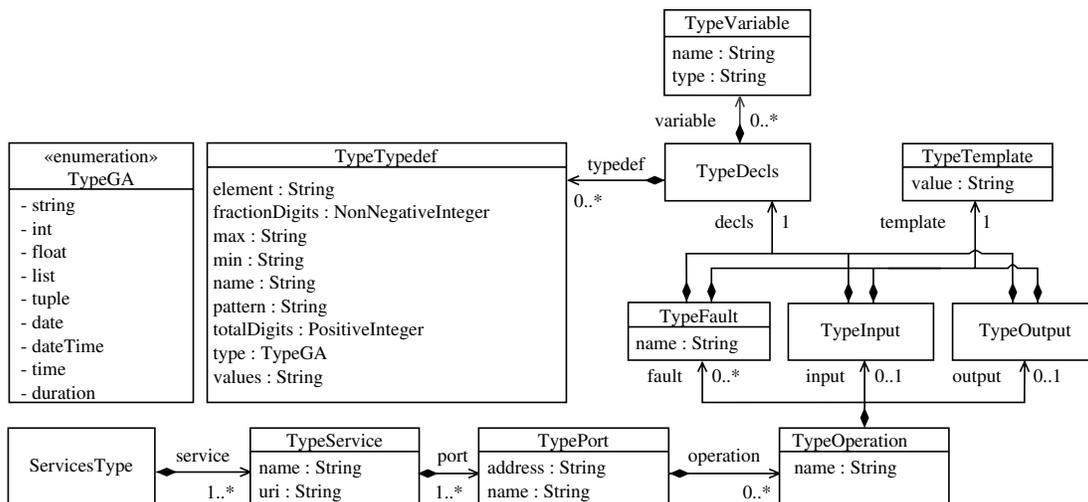


Fig. 9. ServiceAnalyzer service catalogue metamodel

C. Model extraction

Since WSDL documents are declarative and language-independent descriptions of the Web Services, the original proposal intended to use them as design models. After transforming automatically the XML Schema description of the WSDL document format into a regular ECore metamodel [27], WSDL documents would be loaded as regular Eclipse Modeling Framework models, reusing most of the technologies mentioned in Section VI.

In practice, however, WSDL documents are too complex to be used as-is for model weaving and model transformation. WSDL documents can be divided across multiple files and machines and combine descriptions in the WSDL and XML Schema formats. In addition, XML Schema and WSDL are highly flexible, allowing many possibilities that may or may not be implemented by vendors. This has led to the definition of specifications such as the Web Services Interoperability Basic Profile (WS-I BP) [28], which restricts these standards to a consistent subset that is well implemented across vendors.

Therefore, it was decided to extract models from the WSDL documents themselves using a new custom tool, ServiceAnalyzer, also available as open source from [17]. ServiceAnalyzer produces a “service catalogue” from a set of local or remote WSDL documents that conform to the WS-I BP. Service catalogues can be loaded as an EMF model by using their XML Schema definition, as originally intended for WSDL.

The service catalogue metamodel is shown in Figure 9. Models are instances of *ServiceType*, which contains a set of *TypeServices* with their own *TypePorts*. Each *TypePort* has a collection of *TypeOperations* that may have an input, an output, and/or several fault messages. Message descriptions are divided into a *TypeTemplate* containing an Apache Velocity [29] template, and a *TypeDecls* that declares the variables used within the Velocity template. Variables may belong to one of the predefined types in *TypeGA*, which are based on the XML Schema primitive types, or they may belong to a

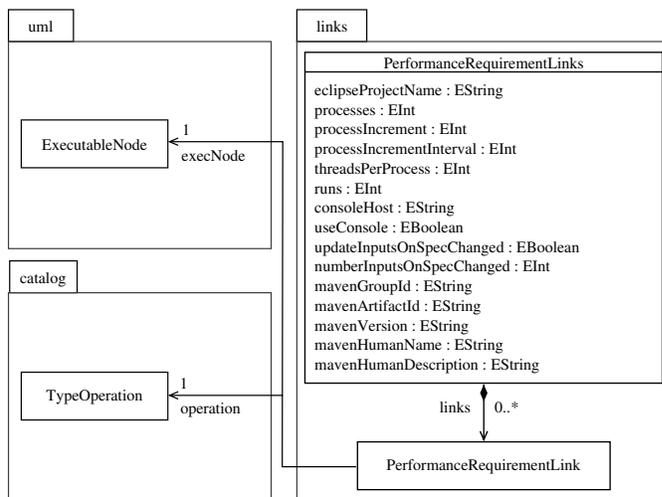


Fig. 10. ServiceAnalyzer-MARTE weaving metamodel

custom type defined with a *TypeTypedef*.

Services store their names and namespace URIs, ports store their names and the URLs they are listening at, and operations and faults store their names. Type definitions must specify at least a name and a base type, but they usually specify additional restrictions such as a pattern based on a regular expression (*pattern*), minimum or maximum values (*min* or *max*) or a set of accepted values, among others.

D. Weaving metamodel

The weaving model needs to relate the *ExecutableNodes* in the UML activity diagram with the *TypeOperations* in the ServiceAnalyzer service catalogue. For instance, a developer might want to ensure that every invocation of the *evaluate* operation of the *Order* service finishes within a certain time while handling a certain number of requests per second.

The weaving metamodel is shown in Figure 10. It is quite similar to that in Figure 9, but the global options in the

PerformanceRequirementLinks class have been changed to reflect the target technologies for this transformation:

- `eclipseProjectName` is the name of the Eclipse project which will be generated by the transformer. By default, it is set to “performance.tests”.
- The attributes ranging from `process` to `useConsole` are directly mapped to the configuration options of The Grinder with the same name. `process` is the number of worker processes that will be used by each agent, starting from 1 and increasing by `processIncrement` every `processIncrementInterval` milliseconds (by default, by 1 every second). Each worker process will spawn as many as `threadsPerProcess` threads and repeat the tests the number of times indicated in `runs`. If `useConsole` is set to “true”, the console process at `consoleHost` will distribute work over the agents connected to it.
- The rest of the attributes can be used to customise the metadata of the Maven project that is generated by the transformer.

As for the options for the testing process itself, `updateInputsOnSpecChanged` and `numberInputsOnSpecChanged` indicate if the test inputs should be updated when the `.spec` file describing their format changes, and how many should be generated each time.

The default options should be good enough for most users. In the next sections, we will mention again some of them as we introduce the following steps in the generation process.

E. Test data generation

In order to run performance tests, it is necessary to provide them with test inputs so they can exercise the WS appropriately. Doing this in a completely automated way is outside the scope of the approach. As an initial approximation, data inputs are randomly generated using uniform distributions, based on the variables and templates in the service catalogue.

First, the tools extract the appropriate Velocity templates and variable declarations from the ServiceAnalyzer service catalogue to separate files.

Listing 4 shows an Apache Velocity template which can produce every valid request for an order evaluation service, according to its WSDL and XML Schema declarations. As a template language, the Velocity language is kept simple, providing only the most common programming constructs, such as conditionals (`#if`), loops over a list (`#foreach`), variable assignments (`#set`) or field references (`$var.field`). Velocity templates are expanded during test execution with the variables loaded into their *contexts*. This template produces a `<newOrder>` element for each item in `$evaluate`. In turn, the template produces a `<articleQuantities>` element for each item, with the appropriate article identifier and requested quantities.

Listing 5 shows the TestSpec declarations that were extracted from the same catalogue entry. The TestSpec language is implemented by the TestGenerator tool, also available

```
<w:evaluate xmlns:w="http://ws.sodmt.uca.es/">
  #foreach($V1 in $evaluate)
    <newOrder>
      #foreach($V2 in $V1)
        <articleQuantities>
          <articleID>
            $V2.get(0)
          </articleID>
          #foreach($V3 in $V2.get(1))
            <quantity>
              $V3
            </quantity>
          #end
        </articleQuantities>
      #end
    </newOrder>
  #end
</w:evaluate>
```

Listing 4. Apache Velocity template extracted from the ServiceAnalyzer catalog for producing the body of a message from test data

```
typedef int (min=0, max=100) TArtID;
typedef float (min=0.01, max=2000) TPrice;
typedef list (element=TPrice, min=1, max=1) TL_float;
typedef tuple (element={TArtID, TL_float}) TArticleQtys;
typedef list (element=TArticleQtys, min=0) TOrder;
typedef list (element=TOrder, min=1, max=1) TEvaluate;
TEvaluate evaluate;
```

Listing 5. TestGenerator `.spec` extracted from the ServiceAnalyzer catalog describing the test data for the template in Listing 4

from [17]. It is a simple domain-specific language (inspired on C declarations) which allows users to define new scalar, list and tuple types based on a set of primitive types based on XML Schema. These new types can have additional constraints, such as having minimum or maximum values or lengths, adhering to a certain regular expression or having a certain number of digits. From these declarations, TestGenerator can produce an arbitrary number of random tests and store them as Velocity templates.

The Velocity files produced by TestGenerator set up the context to be used to generate the message templates. They consist of a sequence of variable assignments in which every variable receives a list of values to be used within each test. Listing 6 shows three test cases that were produced from

```
#set($evaluate = [
  [[[85, [1530.1414]], [3, [1652.419]], [50, [550.96515]]],
  [[[92, [1682.8262]], [45, [1593.5898]]],
  [[[79, [72.64899]], [22, [603.8968]], [8, [1278.9677]]]])
)
```

Listing 6. Test data produced by TestGenerator from the `.spec` in Listing 5

```

grinder.processes=5
grinder.runs=100
grinder.processIncrement=1
grinder.processIncrementInterval=1000

```

Listing 7. Example `.properties` file with configuration parameters for the workload

```

class TestRunner:
    def __call__(self):
        def invoke():
            response = HTTPRequest().POST(
                "http://localhost:8080/orders",
                "(...SOAP_message...)")
            stats = grinder.statistics.getForCurrentTest()
            stats.success = (response.statusCode != 200
                and stats.time < 150)
            test = Test(1, "Query_order_by_ID").wrap(invoke)
            test()

```

Listing 8. Example Jython script for The Grinder with the contents of the performance test to be run by each simulated client

the `.spec` in Listing 5. For instance, the first test requests 1530.14 units of article #85, 1652.419 units of article #3 and 550.965 units of article #50. We used Velocity to store test data since it was more flexible than a simple table or spreadsheet, as it allowed for arbitrarily nested lists.

In the wild, WSDL declarations tend to be quite lax, allowing messages with no upper bound on their length or elements containing negative integers, even though they are not accepted. In these cases, users may want to customise the service catalogue before generating the `.spec` descriptions from it. This will change the values used for all tests of the modified operations. Alternatively, users may want to modify a single `.spec` file describing the inputs of a particular test. Users may also customise the message templates with additional logic, or provide manually designed input data instead of generating random inputs.

The explicit separation of the service interface, message generation template, test generation specification and test data provides a great deal of flexibility. Later iterations of this application could generate larger parts of the test plan by implementing more advanced test generation strategies beyond random generation. These advanced strategies could be expressed as part of the links in the weaving model. The strategy could be applied in the weaving model refining step showed in Figure 3.

F. Test code generation

After weaving the service catalogue model with the MARTE model and producing some input data to exercise the Web Services, the next step is generating the test specification for The Grinder.

The Grinder requires generating two different files: a `.properties` file indicating several parameters of the work-

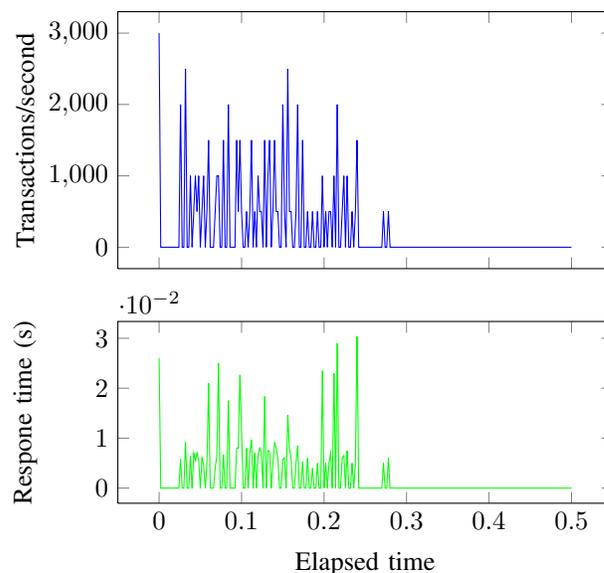


Fig. 11. Overall performance graph produced by Grinder Analyzer

load to be generated, and a Jython script with the test to be run by each simulated client. Listings 7 and 8 show simplified examples for these two files. These files are automatically generated using EGL.

The `.properties` file in Listing 7 indicates that 5 processes should each run the test 100 times, starting with 1 process and adding one more every 1000 milliseconds. On the other hand, the test itself consists of sending an appropriate SOAP message to a specific URL and checking that the response has the OK (200) HTTP status code and that it was received within 150 milliseconds. These values are extracted from the global options in the `PerformanceRequirementLinks` object of the model. `consoleHost` and `useConsole` are also used in the `.properties` file.

The actual generated Jython script is over 180 lines long and takes advantage of several language features to avoid code repetition. In addition to running the tests themselves, it can regenerate test data if the `.spec` files have been customised by the user since the last run. Every time a test is run, a set of input values is randomly selected from the available test data. This input data is used to generate the SOAP message from the message templates, invoke the service and check the non-functional attributes of the reply. One limitation with the current version of the scripts is they can only check maximum response times, unlike the approach in Section VI, which can handle averages, medians and percentiles.

G. Test infrastructure and report generation

The approach in Section VI was straightforward: as it simply produced Java code based on the ContiPerf library, users would simply need to add ContiPerf to their development environments and the run the tests using standard tools. However, running the tests produced by this approach would require setting up TestGenerator, The Grinder, and Apache Velocity.

Operation	Passed tests	Failed tests	Bytes per second	Mean response length
Close Order	60	0	66,590	332.95
Evaluate Order	60	0	64,333.33	321.67

TABLE I
TEST METRICS PRODUCED BY GRINDER ANALYZER (OVERALL RESULTS, THROUGHPUT AND MESSAGE SIZES)

Operation	Mean response time	Response time std. dev.	Mean time DNS	Mean time conn.	Mean time first byte
Close Order	14.35	15.62	0	0.13	13.18
Evaluate Order	8.98	6.01	0	0.37	5.93

TABLE II
TEST METRICS PRODUCED BY GRINDER ANALYZER (TIMING INFORMATION)

For this reason, the tools implement an additional EGL transformation that produces an Apache Maven [30] project description that automatically downloads all dependencies, runs the performance tests and produces test reports from the results. The Grinder is integrated through the open source plug-in available at [31]. Maven also enforces a standard directory layout for all the generated artefacts.

This infrastructure allows users to run the entire testing process with a single `mvn post-integration-test` command, which also invokes the Grinder Analyzer tool [32] on the raw logs to produce an HTML report including (but not limited to) the information shown in Figure 11 and Tables I and II. The report includes both a graph with the response times and transactions per seconds obtained, and a table with more detailed information. The report shows that all tests passed and that the mean response time for the tested service was 8.98 milliseconds. These results are to be expected, since the tool was tested against local Web Services using an in-memory object-relational database. Applying this approach to real-world WS is a future line of work.

VIII. CONCLUSION AND FUTURE WORK

This work has described an overall approach for generating performance test artefacts from the abstract performance models produced by the inference algorithms in [2]. To generate concrete test artefacts while keeping the abstract performance models separated from any design or implementation details, the approach links the performance model to a design or implementation model using an intermediate *weaving model*. If a design or implementation model is not available, it can be extracted from the existing code. The weaving model can be then optionally refined using a model-to-model transformation, and finally transformed into the performance test artefacts with a model-to-text transformation.

The general approach has been validated by applying it on two target technologies. Both approaches have been success-

fully implemented and are freely available under the open source Eclipse Public License at [17].

The first application weaves JUnit test suites with MARTE models and converts all or some of their unit tests into performance test cases, using the ContiPerf library. The implementation model is extracted from the Java code implementing the test cases using the model discovery tool MoDisco [19], and the weaving model links the *ExecutableNodes* in the UML activity diagram to the Java tests in the MoDisco model.

The second application can generate performance test cases for any Web Service that is described using the WSDL specification [21]. It is independent of the language in which the Web Service has been implemented, as it is based on a special-purpose performance testing tool: The Grinder [24]. Users extract service catalogues from a set of WSDL documents and then weave the service operations in the catalogue with the MARTE models. The service catalogues also include message templates and template variable declarations, which are used to randomly generate a set of initial test inputs. Users are able to manually customise the service catalogue, the message templates and the test inputs. In addition to the inputs, a set of automated model-to-text transformation produces the Jython code and the configuration file required by the Grinder, and a Maven project description that enables users to run the tests and produce reports with a single command (as those in Section VII-G).

While these applications show that the overall approach can be reused for different target technologies, they do currently share several limitations. Transformations only know the part of the system under test that is strictly needed to generate the tests. For this reason, users will need to manually customise the tests if they need to restore the state of the system after a performance test, a memory violation or an aborted operating system process, or if they want to set up specific mockups for specific subsystems in the application. Nevertheless, the transformations could assist the user by providing clear “hooks” where this kind of logic could be placed, and keeping those “hooks” from being overwritten if the tests are generated. This is already being done in the Java approach: the generated test suites use EGL protected areas that are preserved when the files are regenerated.

Future lines of work include:

- Continue evaluating both approaches by applying them to larger Web Services running in remote services and using larger data sets. The present versions were developed using local WS with small in-memory databases. One of the case studies under consideration is the Worldtravel testbed in [33], which implements a working business application backed by a relational database with more than 1GB of data.
- Enhance the Jython code generated in the second approach to include the same target metrics as in ContiPerf, such as average time, median time or percentiles. Usually, Service Level Agreements are defined in terms of percentages (“90% of the requests should be attended in x seconds”).

- Provide more advanced strategies for generating test inputs for the WSDL-based WS. Currently, all inputs are generated using a uniform random distribution, but other random distributions could be combined. Alternatively, an evolutionary algorithm could be used to look for test cases that produce SLA violations, as proposed by Di Penta et al. in [9].
- Handle more complex service level agreements beyond meeting a certain service level objective (throughput and/or response times in our case). For instance, the current algorithms do not take into account the fact that a developer may want to enforce different SLOs than a customer in production. These particular variations could be handled by the weaving model itself, however, by adding appropriate global options to scale back the performance requirements.
- Evaluate the overall approach for other target technologies, such as unit tests written for other programming languages or different kinds of systems altogether, such as multimedia applications or graphical user interfaces.

ACKNOWLEDGEMENTS

This work was funded by the research scholarship PU-EPIF-FPI-C 2010-065 of the University of Cádiz, the MoDSOA project (TIN2011-27242) under the National Program for Research, Development and Innovation of the Ministry of Science and Innovation (Spain) and by the PR2011-004 project under the Research Promotion Plan of the University of Cádiz.

REFERENCES

- [1] A. García-Domínguez, I. Medina-Bulo, and M. Marcos-Bárcena, "An approach for performance test artefact generation for multiple technologies from MARTE-Annotated workflows," in *7th International Conference on Internet and Web Applications and Services (ICIW 2012)*, Stuttgart, Germany, June 2012.
- [2] A. García-Domínguez, I. Medina-Bulo, and M. Marcos-Bárcena, "Model-driven design of performance requirements with UML and MARTE," in *Proceedings of the 6th International Conference on Software and Data Technologies*, vol. 2. Seville, Spain: SciTePress, July 2011, pp. 54–63.
- [3] M. Woodside, G. Franks, and D. Petriu, "The future of software performance engineering," in *Proc. of Future of Software Engineering 2007*, 2007, pp. 171–187.
- [4] D. C. Petriu and H. Shen, "Applying the UML Performance Profile: Graph Grammar-based Derivation of LQN Models from UML Specifications," in *Proc. of the 12th Int. Conference on Computer Performance Evaluation: Modelling Techniques and Tools (TOOLS 2002)*, ser. Lecture Notes in Computer Science. London, UK: Springer Berlin, 2002, vol. 2324, pp. 159–177.
- [5] M. Tribastone and S. Gilmore, "Automatic extraction of PEPA performance models from UML activity diagrams annotated with the MARTE profile," in *Proc. of the 7th Int. Workshop on Software and Performance*. Princeton, NJ, USA: ACM, 2008, pp. 67–78.
- [6] Object Management Group, "UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) 1.0," <http://www.omg.org/spec/MARTE/1.0/>, November 2009, last checked on 2012-03-03.
- [7] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing," Working Paper 04/2006, April 2006, last checked on 2012-12-20. [Online]. Available: <http://researchcommons.waikato.ac.nz/handle/10289/81>
- [8] C. Barna, M. Litoiu, and H. Ghanbari, "Model-based performance testing (NIER track)," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11. New York, NY, USA: ACM, 2011, pp. 872–875.
- [9] M. Di Penta, G. Canfora, G. Esposito, V. Mazza, and M. Bruno, "Search-based testing of service level agreements," in *Proceedings of Genetic and Evolutionary Computation Conference*, H. Lipsch, Ed. London, United Kingdom: ACM, July 2007, pp. 1090–1097.
- [10] K. Suzuki, T. Higashino, A. Ulrich, T. Hasegawa, A. Bertolino, G. De Angelis, L. Frantzen, and A. Polini, "Model-based generation of testbeds for web services," in *Testing of Software and Communicating Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, vol. 5047, pp. 266–282.
- [11] M. D. Del Fabro, J. Bézin, and P. Valduriez, "Weaving models with the Eclipse AMW plugin," in *Proceedings of the 2006 Eclipse Modeling Symposium, Eclipse Summit Europe*, Esslingen, Germany, October 2006.
- [12] J. M. Vara, M. V. De Castro, M. Didonet Del Fabro, and E. Marcos, "Using weaving models to automate model-driven web engineering proposals," *International Journal of Computer Applications in Technology*, vol. 39, no. 4, pp. 245–252, 2010.
- [13] OMG, "UML Profile for Schedulability, Performance, and Time (SPTP) 1.1," January 2005, last checked on 2012-12-20. [Online]. Available: <http://www.omg.org/spec/SPTP/1.1/>
- [14] —, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (QFTP) 1.1," <http://www.omg.org/spec/QFTP/1.1/>, April 2008.
- [15] D. S. Kolovos, "Epsilon ModeLink," 2010, last checked on 2012-12-20. [Online]. Available: <http://eclipse.org/epsilon/doc/modelink/>
- [16] D. S. Kolovos, R. F. Paige, L. M. Rose, and A. García-Domínguez, "The Epsilon Book," 2011, last checked on 2012-12-20. [Online]. Available: <http://www.eclipse.org/epsilon/doc/book>
- [17] A. García-Domínguez, "Homepage of the SODM+T project," January 2012, last checked on 2012-12-20. [Online]. Available: <https://neptuno.uca.es/redmine/projects/sodmt>
- [18] V. Bergmann, "ContiPerf 2," September 2011, last checked on 2012-12-20. [Online]. Available: <http://databene.org/contiperf.html>
- [19] H. Bruneliere, J. Cabot, F. Jouault, and F. Madiot, "MoDisco: a generic and extensible framework for model driven reverse engineering," in *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, Antwerp, Belgium, September 2010, pp. 173–174.
- [20] H. Haas and A. Brown, "Web services glossary," World Wide Web Consortium, W3C Working Group Note, February 2004, last checked on 2012-12-20. [Online]. Available: <http://www.w3.org/TR/ws-gloss/>
- [21] World Wide Web Consortium, "WSDL 2.0 part 1: Core Language," June 2007, last checked on 2012-12-20. [Online]. Available: <http://www.w3.org/TR/wsdl20>
- [22] Apache Software Foundation, "Apache CXF," November 2012, last checked on 2012-12-20. [Online]. Available: <https://cxf.apache.org/>
- [23] Java.net, "JAX-WS reference implementation," November 2011, last checked on 2012-12-20. [Online]. Available: <http://jax-ws.java.net/>
- [24] P. Aston and C. Fitzgerald, "The Grinder, a Java Load Testing Framework," 2012, last checked on 2012-12-20. [Online]. Available: <http://grinder.sourceforge.net/>
- [25] Apache Software Foundation, "Apache JMeter," November 2011, last checked on 2012-12-20. [Online]. Available: <http://jakarta.apache.org/jmeter/>
- [26] eviware.com, "loadUI homepage," 2012, last checked on 2012-12-20. [Online]. Available: <http://www.loadui.org/>
- [27] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework*, 2nd ed., ser. Eclipse Series. Addison-Wesley Professional, December 2008.
- [28] Web Services Interoperability Organization, "Basic profile - version 1.1 (Final)," April 2006, last checked on 2012-12-20. [Online]. Available: <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>
- [29] Apache Software Foundation, "Apache Velocity Project homepage," November 2010, last checked on 2012-12-20. [Online]. Available: <http://velocity.apache.org>
- [30] —, "Apache Maven homepage," January 2012, last checked on 2012-12-20. [Online]. Available: <http://maven.apache.org>
- [31] G. Iacono and F. Muñoz-Castillo, "grinder-maven-plugin homepage," July 2012. [Online]. Available: <http://code.google.com/p/grinder-maven-plugin/>
- [32] T. Bear, "Grinder Analyzer homepage," July 2012, last checked on 2012-12-20. [Online]. Available: <http://track.sourceforge.net/>
- [33] P. Budny, S. Govindharaj, and K. Schwan, "Worldtrawl: A testbed for service-oriented applications," *Service-Oriented Computing/ICSO 2008*, pp. 438–452, 2008.

A Cross-Domain Query Navigation and Visualization System for Touchscreens that Exploits Social Search History

Shuichi Kurabayashi

Faculty of Environment and Information Studies

Keio University

5322 Endo, Fujisawa, Kanagawa 252-0882, Japan

kurabaya@sfc.keio.ac.jp

Ryo Shimaoka

Faculty of Policy Management Studies

Keio University

5322 Endo, Fujisawa, Kanagawa 252-0882, Japan

s09411rs@sfc.keio.ac.jp

Abstract—Tablets and smartphones have become the most popular Internet client system for end-users. However, conventional web search engines employ Hyper Text Markup Language (HTML) text input systems that require many individual characters. These are unsuitable for mobile terminals, which are normally equipped with touchscreens. We propose a cross-domain query navigation and visualization system that assists the query input process by providing a context-dependent *word map* that presents the relevance between keywords. This *word map* enables both a narrowing action, whereby users append a new keyword to specify the context of a query, and a sliding action, whereby users replace a keyword to change the query context. The *word map* is unique in that it recommends queries for both narrowing and sliding transitions by computing the directional relevance between the input keyword and another keyword in the social knowledge base. The system is applicable to existing search engine query logs, social networking services, and web browsers, enabling users to control the term recommendations by selecting the logs to be analyzed.

Keywords - Query Navigation; Personalization; User Interface; Collective Intelligence; Web Search Engine.

I. INTRODUCTION

This paper describes a cross-domain query navigation and visualization system [1] and its implementation framework towards modern smart devices, such as smartphones and tablets. The name of our system is *Query Map (Q-MAP)*, and it assists in the input of multiple queries for web search engines by showing a context-dependent *word map* to present the relevance between keywords by considering the current user's input. We implement this system on top of an existing web search engine by utilizing modern HTML5 technologies.

Recent years have witnessed a rapid rise in the popularity of tablet devices and smartphones, and concomitantly, a widespread increase in the use of touch-based user interfaces (UIs). Statistics published by Cisco indicate that global mobile data traffic grew 2.6-fold in 2010, nearly tripling for the third consecutive year [2]. In addition, statistics published by Google Confidential and Proprietary suggest that, by 2015, more than a quarter of mobile traffic will be used for information retrieval and the number of Internet users not using personal computers (PCs) will increase to 788 million [3]. Hence, a major shift in the type of Internet-connected devices, from PCs to mobile terminals, is currently underway.

A large portion of Internet activity is in the form of queries to search engines. However, many users have difficulty querying a search engine on a complex topic that encompasses several terms, such as “JavaScript and HTML5” or “ActionScript and

Application Program Interface (API),” relating to a subject with which they are not familiar. Mobile devices present an additional difficulty: although touchscreens are generally very convenient, they are not particularly well adapted for use as a typing tool. In particular, queries in Chinese, Japanese, Korean, and Vietnamese (CJKV) present special difficulties because each CJKV character requires two or three input strokes. In mobile devices, predictive methods are the predominant means of supporting the input of long sentences and terms. These predictive input methods recommend terms and sentences that can be concatenated to the user's input character sequence. Another conventional method is a keyword suggestion approach, such as Google Suggest. When a user inputs an initial query term, this method suggests related terms by calculating the inter-term relevance, exploiting the search engine's query log to recognize the relevant terms. However, these conventional methods are based on co-occurrence probabilities, and are thus unsuitable for inputting queries that consist of several cross-domain terms, such as “climbing healthcare costs.” In such cases, predictive input methods may not correctly recommend the next search term, and a cross-domain term-relevance calculation is required. Thus, the UIs provided by conventional web search engines require users to tap the keyboard or screen many times, making them unsuitable for mobile terminals.

This paper proposes a cross-domain query navigation system that assists in the input of multiple queries by forming a content-dependent *word map* to present the relevance between keywords. This system allows users to select appropriate keywords in a convenient manner, because the word map shows the next coordinate instantly, as shown in Figure 1. In this paper, we extend the system proposed in [1] to support a large-scale dataset and perform practical experiments to evaluate our model. In addition, this paper shows a new prototype system implementation running on the iPad and Android smart devices. This implementation utilizes modern touchscreens as a user interface for inputting queries with small number of touch operations.

This *word map* approach makes it possible to reduce the number of keyboard or screen taps. For example, when a user wishes to add the search terms “global,” “mobile,” and “traffic” to the term “statistics,” which has already been inserted in the search box, only one tap is required for each term, making three in all. The keywords are presented after considering the user's browser history, which enables personalization, and other users' querying history, which supports users by exploiting collective intelligence. Our system configures the balance between personalization and collective intelligence support dynamically, which is not possible with conventional search engines. This configuration mechanism can be applied to protect users' privacy by setting all the query logs to be stored in a client local storage.

Another advantage of our system is its applicability to the search histories of social networks, which include groups of

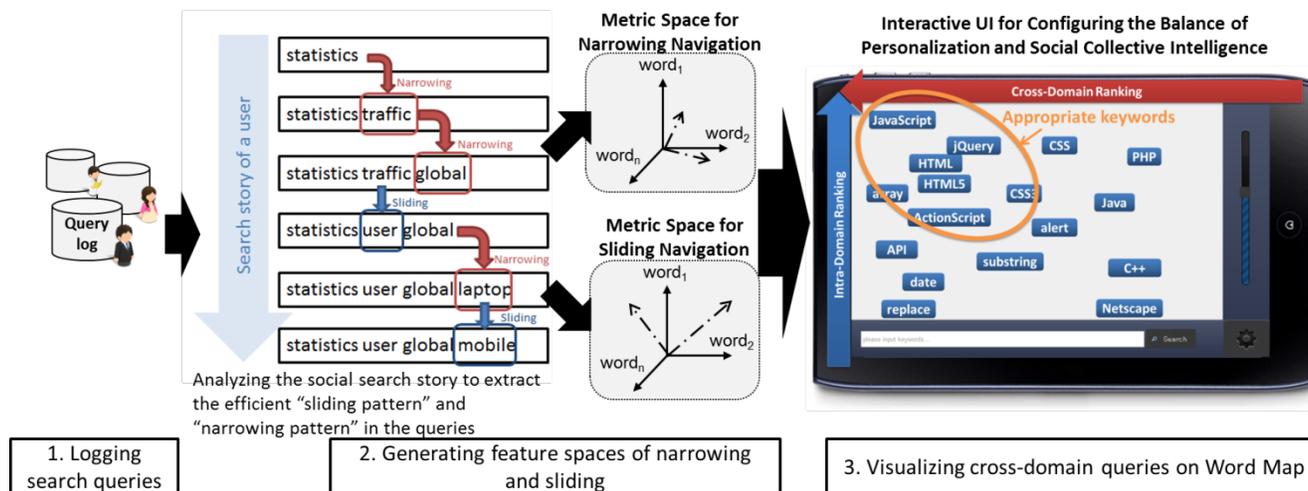


Figure 1. User Interface of a Cross-Domain Query Navigation System.

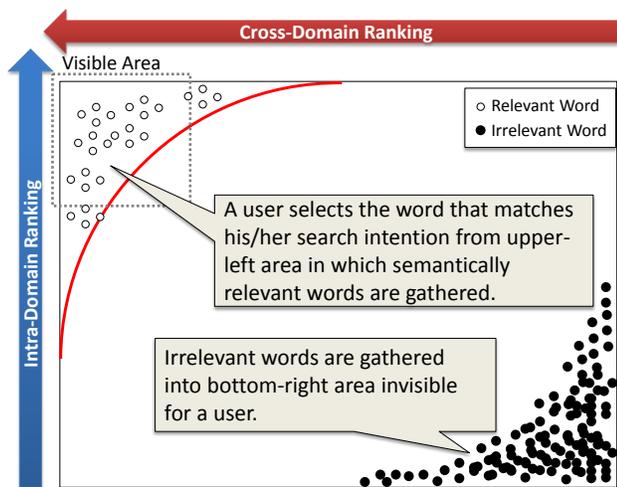


Figure 2. Visual Metric Space for Selecting Relevant Words.

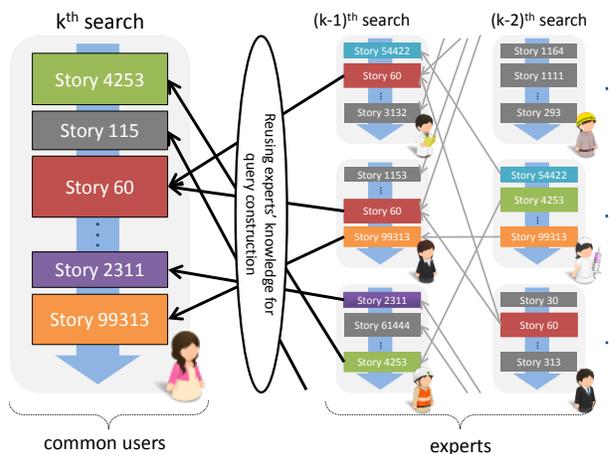


Figure 3. Search Story Sharing among Users Empowers the System's Cross-Domain Keyword Recommendation.

experts in various fields, as shown in Figure 3. This allows users to search within a domain that they are not familiar with by drawing on the collective knowledge and experience of expert groups through their search stories. Furthermore, the application can also help users to construct a query in a language that they do not know well.

The remainder of the paper is structured as follows. Section II demonstrates several advantages of our system. In Section III, we discuss several related studies. and Section IV presents an architectural overview of our system. Section V introduces the prototype system, which is then evaluated in Section VI. Finally, Section VII gives our concluding remarks and some ideas for future work.

II. ADVANTAGES OF CROSS-DOMAIN QUERY NAVIGATION

Here, we explain the example scenario of query navigation shown in the center of Figure 1. This figure shows the following types of navigation:

- **Narrowing:** Users append a new keyword (e.g., “traffic,” or “global”) to specify the context of a query. The appended keyword is at a lower level of abstraction than those of the existing keywords.

- **Sliding:** Users replace a keyword to change the context of a query. Here, the user removes an existing keyword (e.g., “traffic”) that is not within the scope of the current topic of interest and inserts another one (e.g., “user”) that is relevant to the current topic of interest, thus shifting the focus of the query. In this case, the system recommends a new keyword (e.g., “laptop”) as being appropriate in the current context.

The advantage of this system is that it obviates the need for users to enter subsequent search terms themselves; instead, they are able to select from among those that are mapped on the screen. Figure 2 shows a metric space for visualizing the narrowing and sliding navigation methods. Our system computes the semantic distance between words for each navigation type and allocates each word to the two-dimensional map according to the calculated distance. The most important factor of this visualization is that it eliminates irrelevant words from the visible upper-left area, because the system orders the retrieved words by using the correlation scores between each word. The correlation-based distance enables us to locate irrelevant words into bottom-right corner. This combination mechanism of visualization and correlation computation is our unique approach. The user is able

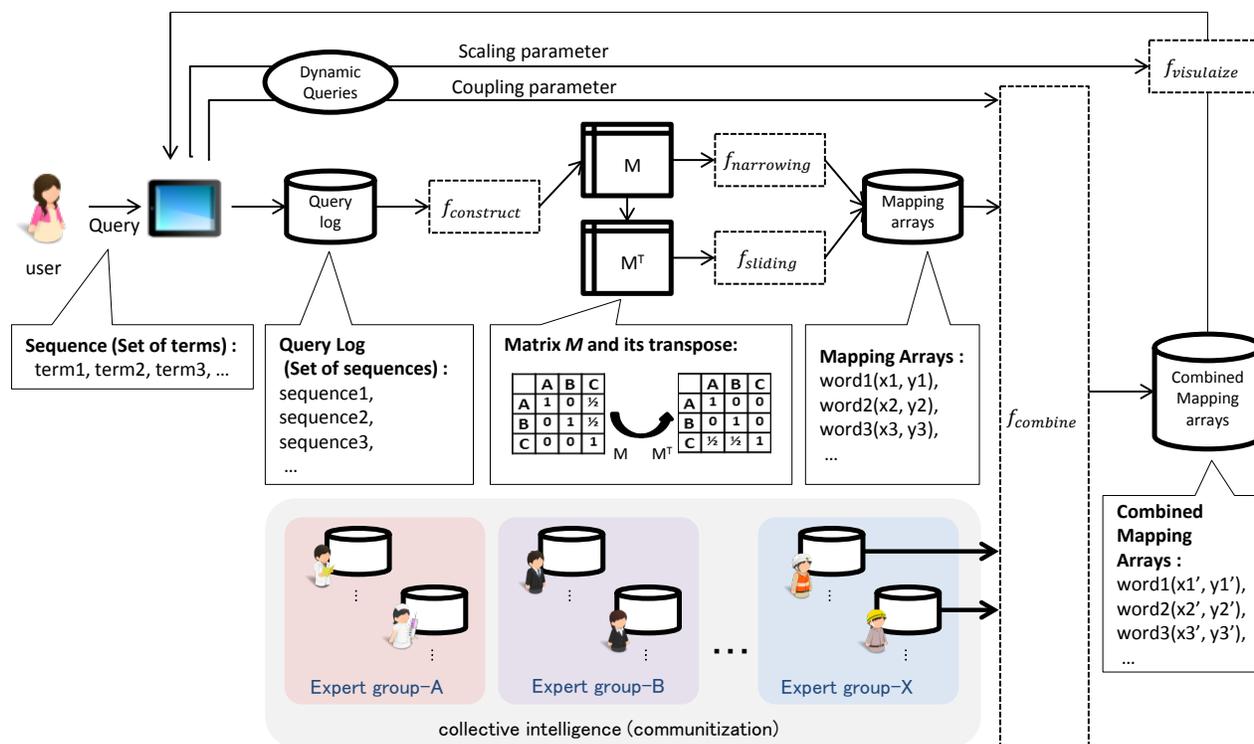


Figure 4. System Architecture for the Recommendation of Search Terms on Word Map for Directional Relevance between Input Keyword and Another Keyword in the Log.

to search something for making trial and error while watching the change of search results. The horizontal axis corresponds to the cross-domain ranking where the keywords are ordered according to the sliding navigation. The vertical axis corresponds to the intra-domain ranking where the keywords are ordered according to the narrowing navigation. Users can select the words that match their search intention from the upper-left area in which semantically relevant words are clustered. In contrast, irrelevant words are mapped to the bottom-right.

This system utilizes the query logs, which are submitted by experts, as a knowledge base for providing search navigations. This application reuses the experts' query as a successful search story. For example, a novice user submits a query "asthma, how-to cure." By using the conventional search method, the novice user has difficulty finding the sufficient information. On the other hand, our system reuses the query logs, submitted by experts who know the field well, to navigate a user to use "asthmatic remission".

Our navigation algorithm is independent of a search engine's relevance computation method such as PageRank. This is because our system calculates context-dependent relevance between query words by analyzing co-occurrence of those words in a query and temporal relevance between words in a sequence of queries submitted for a specific purpose. Thus, our system can be implemented as a meta-level system or a wrapper interface, which is installed in modern smart devices including smart TV and the latest gaming console, for the legacy search engines such as Google and Microsoft's Bing. This is highly advantageous for realizing a novel UI suitable for touchscreens because we do not have to re-implement or to modify the existing Web search engine infrastructure. It is valuable to mention that our system

can be a special-purpose search UI for the specific site. For example, by applying our system to an online bookstore, our system will navigate users to input book titles, authors, and publisher's name.

III. RELATED WORK

The query expansion method is a well-known means of helping a search engine's users to input complex queries [4]. The traditional example of query expansion is Google Suggest, which recommends keywords by analyzing query logs based on the number of previous searches. The significant difference between the conventional keyword suggesting system and our Q-MAP system is a smart visualization mechanism that integrates query logs analysis and intuitive noise reduction mechanism for eliminating irrelevant words from the visible area. Our system provides a novel UI designed for modern touchscreen devices. Kelly, Gyllstrom, and Bailey [5] proposed the combination method of term suggestion that helps users to add terms to their original query to clarify the semantics of the initial input. Specialization and parallel movement query suggestion (SParQS) [6] is a query suggestion system that provides two query reformulation methods: a specialization method to make the query more specific, and a parallel movement method to change an entity contained in the query. These approaches utilize a query log collected by a web search engine. Wen et al. [7] proposed a query clustering method based on content similarity for detecting frequently asked queries.

Currently, many researchers are focusing on personalization mechanisms in query expansion [8]. For example, Teevan et al. [9] proposed a personalization method that considers the user's specific interests by constructing a user profile from the relevance feedback in a ranking. Gauch et al. [10] developed an

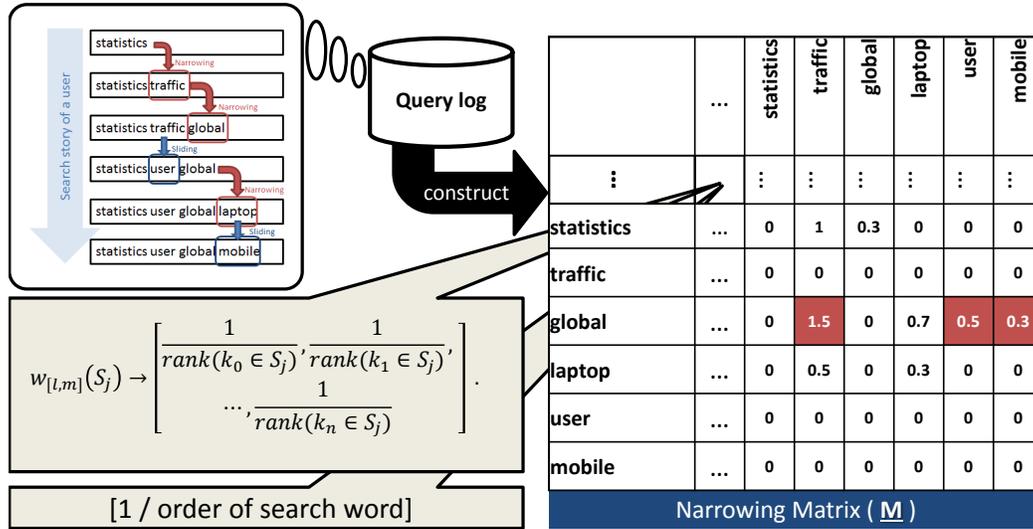


Figure 5. Composition Process of Narrowing Matrix.

implicit personalization mechanism that generates ontology-based user profiles without user feedback by monitoring the user's browsing activities.

An alternative method of query expansion uses the concept of *community*. Smyth et al. [11] introduced the collaborative filtering method, which exploits a similar relationship between queries and results for each community. The method expands a query by referring to a graded mapping between users and items.

In the development of search UIs, many approaches have analyzed the user's search activities. For example, classified or faceted search results are well-known techniques for organizing expanded queries [12], [13].

The most significant difference between our approach and those listed above is context-dependent reuse of query stories shared in social networking service (SNS). Our system focuses on two *dimensions* in the query building process: *narrowing and sliding*. Narrowing is a typical query building method that allows users to increase the specificity of a query after starting with an initial keyword. Our approach also supports sliding, which suggests cross-domain keywords by computing the implicit relevance of keywords in different domains, such as "climbing," "healthcare," and "costs." Unique feature of our system is a method to increase the precision of the sliding and narrowing query specification process by exploiting the search history of a relevant group or community.

IV. CROSS-DOMAIN QUERY NAVIGATION APPROACH

The narrowing and sliding forms of navigation are based on an *inter-term relationship matrix* constructed from a query log, as shown in Figure 4. The purpose of this matrix is to record the relationship between keywords for each user. The system converts the matrix into recommendation scores, which correspond to the coordinate values for narrowing and sliding as presented on the UI. The system combines the recommendation scores from the user with those from social groups within the domain of interest.

The first stage is for the system to construct the matrix from the query log, which is the set of keyword sequences recorded when the user inputs a complex query in a search box. This matrix contains scores representing relationships between search terms. It is updated from the query log. In the second stage, the

system converts the matrix into two recommendation scores: one for sliding and the other for narrowing. The system calculates these scores based on the inner product of the matrix and its transpose. The final stage is to combine the recommendation score of the user with those of social groups within the domain of interest. Our concept of computing social network-based relevance is the reuse of third-party knowledge about query construction. This system may distinguish between several groups of users according to social graphs, such as Twitter's follower/followee structure and Facebook's friend structure. Users can also adjust the parameters of the combination process.

A. Data Structure

The data structure in this system consists of two elements—a *query log* and an *inter-term relationship matrix*—that are now explained in detail.

1) Query Log

A query log is a set of sequences that consist of search terms. We define a *Log (L)* as a data structure based on a *Sequence (S)* of keywords inputted by a user. Log L_i of i -th user is defined by the following equation:

$$L_i := \langle S_0, S_1, \dots, S_n \rangle \quad (1)$$

where m is the number of sequences.

A sequence is a set of searched keywords. Therefore, we define a *Sequence (S)* as a data structure based on *keywords (k)*. Sequence S_j is defined by the following equation:

$$S_j := \langle k_0, k_1, \dots, k_m \rangle \quad (2)$$

where n is the number of keywords.

2) Inter-Term Relationship Matrix

We generate a relationship matrix from the query log. The relationship matrix contains a set of values that represents the directional relevance between each pair of keywords (the *weight* of the association). This is a square matrix whose rows and columns each correspond to the same set of keywords. We define the *Matrix (M)* of user i based on the *weight (w)*.

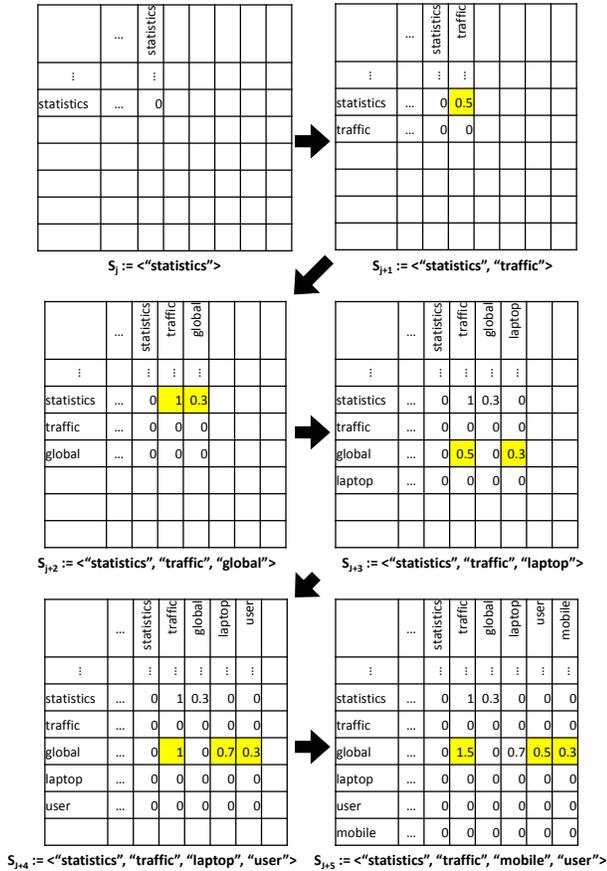


Figure 6. Example of Matrix Composition Process for Cross-Domain Query Navigation.

$$M_i := \begin{bmatrix} W_{[0,0]} & \cdots & W_{[0,q]} \\ \vdots & \ddots & \vdots \\ W_{[q,0]} & \cdots & W_{[q,q]} \end{bmatrix} \quad (3)$$

where q is the number of keywords. The system also generates the matrix transpose M_i^T for reverse look-up.

B. Primitive Functions

The proposed system provides three main functions. The first constructs the relationship matrix from a query log. The second converts the matrix into narrowing and sliding scores for recommendations. The final function combines the recommendation scores of the user with those of a social group that can provide expertise concerning the user's domains of interest.

1) Constructing a Matrix from a Query Log

The system provides a fundamental function to construct a matrix from a query log. The function is defined as follows:

$$f_{construct}(L_i) \rightarrow M_i \quad (4)$$

where M_i is a matrix for i -th user and contains a set of values $w_{[l,m]}$ representing the weights of the directional relevance between k_l and k_m .

This function updates the matrix every time the user inputs a query. Thus, we set the *weight* w of sequence S_j as the relevance based on the *rank* of keyword k .

$$w_{[l,u]}(S_j) \rightarrow \left[\frac{1}{rank(k_0 \in S_j)}, \frac{1}{rank(k_1 \in S_j)}, \dots, \frac{1}{rank(k_n \in S_j)} \right] \quad (5)$$

Figure 5 shows an example of this summation process.

2) Converting a Matrix into Recommendation Scores

The system provides a fundamental function to convert a matrix into mapping arrays. Each mapping array contains the vertical and horizontal scores of a given keyword in relation to the *origin keyword* (z), i.e., the last term of a query. Thus, the function f_{map} generates sliding and narrowing relevance scores according to a keyword specified as the origin point. We define $f_{map}(M_i, a)$ that inputs an origin word a and a user matrix M_i as follows:

$$f_{map}(M_i, z) \rightarrow \{ \langle p_v, p_h \rangle_0, \dots, \langle p_v, p_h \rangle_q \} \quad (6)$$

$$\langle p_v, p_h \rangle_k \rightarrow \left\langle \sum_{x=0}^q M_{i[a,x]} \cdot M_{i[x,k]}, \sum_{x=0}^q M_{i[a,x]}^T \cdot M_{i[x,k]}^T \right\rangle \quad (7)$$

where p_v and p_h are the vertical and horizontal scores, respectively, for the word map. Thus, $\langle p_v, p_h \rangle_k$ corresponds to relevance of k -th word. The vertical score corresponds to the directional relevance of a narrowing search, whereas the horizontal score corresponds to the directional relevance of a sliding search. $M_{i[g,j]}$ denotes the value at a point of g -th row and j -th column.

3) Combining the Recommendation Scores of the User and the Expert Groups

This system uses the collective expertise of other users for its recommendations. This recommendation function merges the user matrix with those of other search engine users in a weighted combination. The system user sets the combination weighting (or *rate*) via a slider on the web page. Thus, we define $f_{combine}$ based on a *combination rate* (r).

$$f_{combine}(p, G, r) \rightarrow \left[\frac{p \cdot (100 - r) + \frac{\sum_{y=0}^e G}{e} \cdot r}{100}, \dots \right] \quad (8)$$

where p is a correlation score and e is the number of people in a *group* (G). These equations combine the matrix of the main user with the average matrix of all users to yield a final score.

C. Query Navigation Methods using Primitive Functions

As shown in Figure 4, our system executes the cross domain query navigation by applying the following seven steps.

Step 1: The system receives an initial query keyword from a user. The initial keyword is required to start our query navigation. We have been recognizing several novices have difficulties to input an initial keyword. In this case, a novice user can combine voice recognition method installed in the modern smartphones and tablets.

Step 2: The system applies $f_{construct}$ primitive function to generate the query matrix M and the corresponding transposed matrix M^T .

Step 3: The system applies f_{map} primitive function, which internally invokes the function $f_{narrowing}$ and the function $f_{sliding}$, into the matrix M and the transposed matrix M^T in order to generate the mapping arrays.

Table 1. Narrowing and Sliding Keywords and their Ranks.

design			e-book			editorial		
rank	narrowing	sliding	rank	narrowing	sliding	rank	narrowing	sliding
1	editorial	e-book	1	design	editorial	1	color	e-book
2	layout	editorial	2	editorial	research	2	design	design
3	color	research	3	color	implication	3	e-book	research
4	image	history	4	layout	history	4	layout	implication
5	scheme	magazine	5	image	program	5	magazine	history
6	research	book	6	scheme	genre	6	newspaper	program
7	ranking	program	7	research	magazine	7	history	genre
8	magazine	genre	8	ranking	book	8	electronic	magazine
9	newspaper	retrieval	9	search	retrieval	9	television	ranking
10	iPhone	iTV	10	engine	brief	10	iPhone	brief

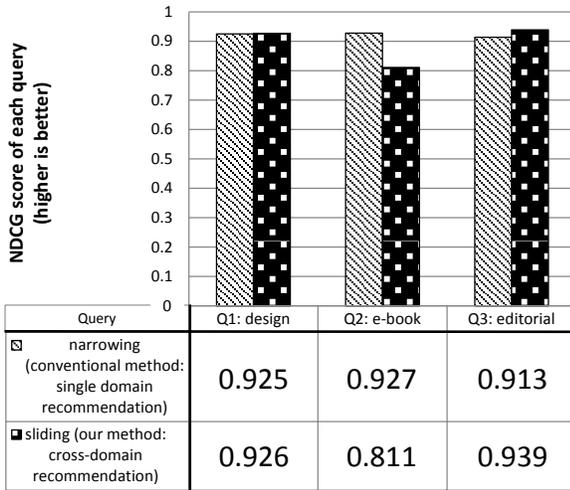


Figure 9. NDCG of Narrowing and Sliding Recommendations.

VI. EXPERIMENTS

This section evaluates the effectiveness of our Q-MAP system when applied to an existing search engine. For this, we implement the Q-MAP prototype system as a Google interface. Two evaluation experiments are performed: Experiment 1 evaluates the precision of the narrowing and sliding relevance computations, and Experiment 2 evaluates the effectiveness of our system by measuring the precision and recall of query expansion. In both experiments, we conduct multiple tests to create a suitable data set.

A. Experiment 1: Overview

Experiment 1 investigates the f_{map} function for converting a matrix into recommendation scores. The experiment evaluates the directional relevance between the input search terms and the candidate search terms, based on the user and community query logs. We compare narrowing, which is a legacy keyword recommendation, and sliding, which is an original feature of this system. We set up the inter-term relationship matrix by submitting 952 queries to Google. As a result, we obtain a 108×108 matrix. The three test topics “design,” “e-book,” and “editorial” are chosen as the initial keywords. These test cases generate three rankings, and we select the top ten keywords for the narrowing and sliding relevancy from each search topic, as shown in Table 1.

This experiment clarifies that our approach calculates the appropriate distance between query keywords. We ask ten test subjects (seven male and three female) to evaluate the relevance

of the 60 keywords from the viewpoint of *narrowing* and *sliding*. Keywords are rated according to the following five-point scheme: 0 (completely irrelevant), 1 (irrelevant), 2 (slightly relevant), 3 (relevant), and 4 (very relevant). We consider the ideal ranking as the average of ten results.

B. Experiment 1: Evaluation Result

To evaluate this experiment, we compute the normalized discounted cumulative gain (NDCG).

$$DCG = \sum_{i=1}^{10} \frac{rel_i}{\log_2 i} \quad (9)$$

$$IDCG = \sum_{i=1}^{10} \frac{rel'_i}{\log_2 i} \quad (10)$$

$$NDCG = \frac{DCG}{IDCG} \quad (11)$$

where rel_i are the average evaluation scores given by the test subjects, and rel'_i are the average scores in descending order.

Figure 8 shows the NDCG of the narrowing and sliding recommendations for our three topics (“design,” “e-book,” and “editorial”). A higher score implies a better retrieval precision. The most important result is the *sliding recommendation* score, because the narrowing recommendation is similar to conventional query methods. The NDCG of the sliding recommendation is almost the same as that of the narrowing recommendation for each query. This implies that our sliding recommendation achieves highly practical precision, although it generates different keywords from the narrowing recommendation. Using this system, users received precise query keywords that shared a cross-domain relationship with the initial keyword. This recommendation is a very powerful tool for inputting a complex query consisting of cross-domain keywords.

C. Experiment 2: Overview

Experiment 2 considers the visualization function that displays the word map according to recommendation scores. The experiment evaluates the precision and recall of the upper-left area of the word map showing the relevant query words. We ask five test subjects (two male and three female) to evaluate the word map, and plot the change in precision and recall while expanding the visible area. By plotting this information, we can evaluate the appropriate size of the visible area to satisfy the user requirements for query expansion precision. The precision p and recall r with a visible area size of k are calculated as follows:

$$p(k) = \frac{V(k) \cap C}{V(k)} \quad (12)$$

$$r(k) = \frac{V(k) \cap C}{C} \quad (13)$$

where $V(k)$ denotes the words displayed in area k and C denotes the set of correct words. The visible area size k is the rectangle containing the top- k narrowing words and the top- k sliding words. As preprocessing for this experiment, we ask five subjects to perform several web searches to find as much information as

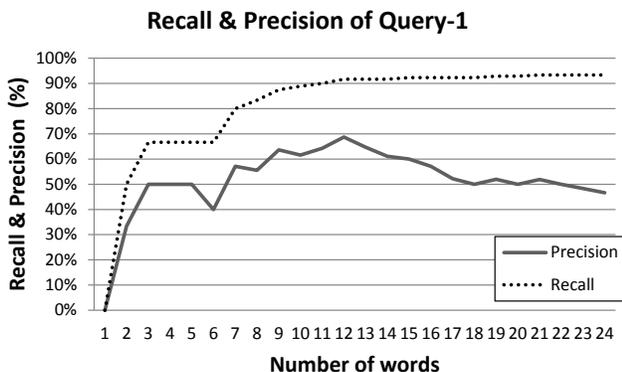


Figure 10. Recall & Precision of Query-1. Initial Keyword is “search.”

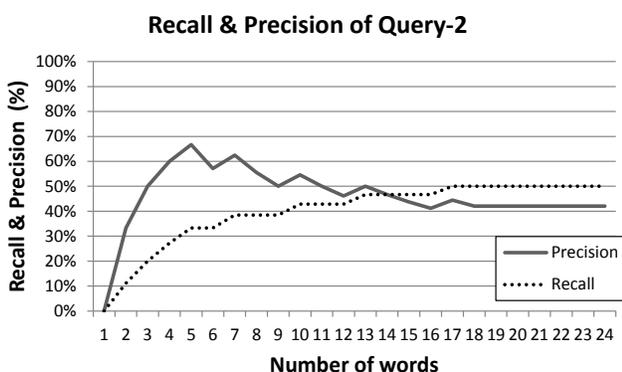


Figure 11. Recall & Precision of Query-2. Initial Keyword is “retrieval.”

possible related to the article [1]. We then ask five different test subjects to perform the same search. We set up the inter-term relationship matrix by submitting 808 queries, which contain 93 keyword variations, to Google. As a result, the system has updated totally 34,865 cells in a 108×108 matrix.

D. Experiment 2: Evaluation Result

We have plotted eight evaluation results corresponding to the initial query keywords “search,” “retrieval,” “interactive,” “words,” “tablet,” “smartphone,” “typing,” and “input” (Figures 10–17). As shown by these graphs, our approach is highly effective at recommending cross-domain keywords when users can select an appropriate initial keyword (Figure 10, Figures 14–17). In contrast, our system could not recommend appropriate keywords when users select an initial keyword that is too common (Figure 11). Figure 12 and 13 show that our system requires sufficiently large query logs to compute the relationship between the initial keywords and other keywords.

Figure 10 shows the recall and precision of Query-1, in which “search” was the initial keyword. In this case, the recall is increasing towards 100% as the number of words k increases. In contrast, the precision has saturated after reaching the top-12 ranking.

Figure 11 shows the results for Query-2 (initial keyword: “retrieval”). This result is not so good because the precision score has saturated beyond the top-5 ranking. This is because the query log does not contain a search history that starts with the keyword “retrieval”.

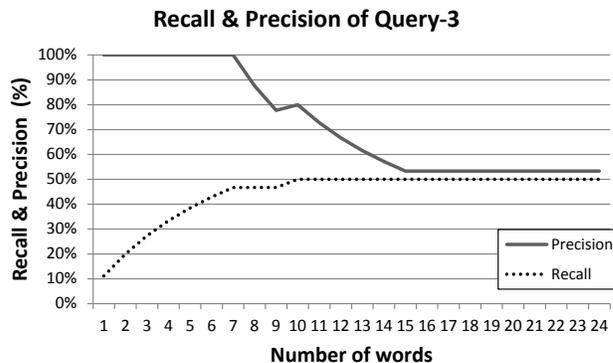


Figure 12. Recall & Precision of Query-3. Initial Keyword is “interactive.”

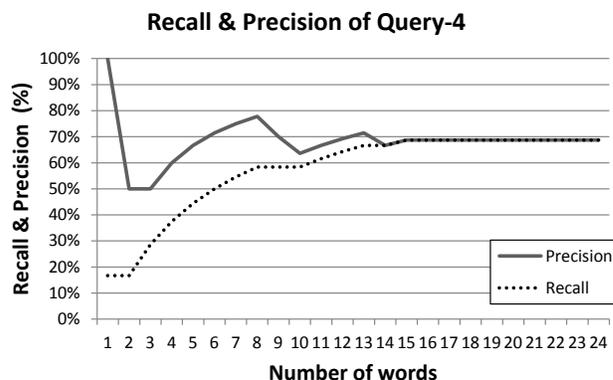


Figure 13. Recall & Precision of Query-4. Initial Keyword is “words.”

Figure 12 and 13 illustrate the limitation of our approach. Figure 12 corresponds to the initial keyword “interactive” and Figure 13 corresponds to the initial keyword “words.” In both cases, our system could not compute the relationship between the initial keywords and other keywords due to the small log size. We recognize that our system requires a sufficient log size in order to accept a wide variety of initial keywords. Currently, our experimental data set is relatively small.

Figures 14–17 show that our system provides good keywords for the cross-domain relevance computation. Specifically, Figure 14 shows the recall and precision for Query-5 (initial keyword: “tablet”). In this case, the precision score remains high while the recall score is increasing. It is important to mention that the reason why the precision score remains high is that the system does not show irrelevant keywords on the screen, instead gathering them into the bottom left area of the metric space, which is not visible to users. Figure 15 shows the recall and precision for Query-6 (initial keyword: “smartphone”). The results are similar to those in Figure 14, but the precision score decreases as more words are included. This is because several irrelevant words move to the center of the metric space. Our metric space may include some irrelevant words in the central area.

We consider this to be a trivial matter because many users do not scroll the metric space or look at the central area. Figure 16 and 17 exhibit ideal results from our system.

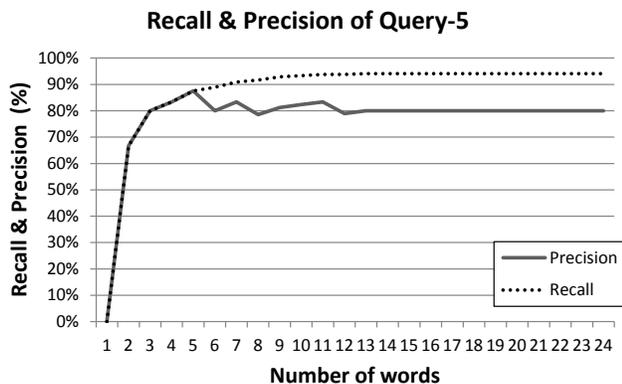


Figure 14. Recall & Precision of Query-5. Initial Keyword is “tablet.”

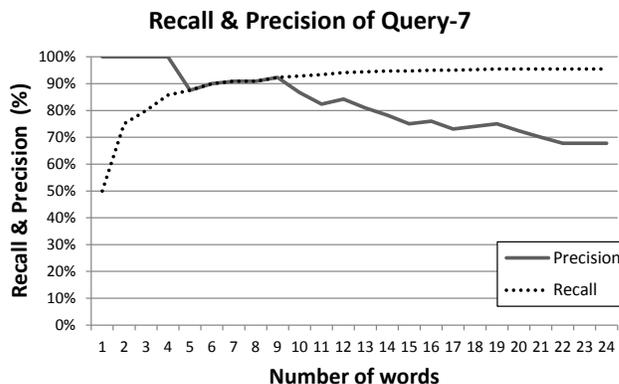


Figure 16. Recall & Precision of Query-7. Initial Keyword is “typing.”

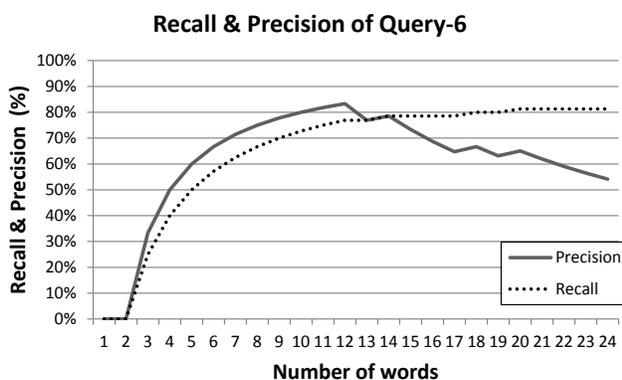


Figure 15. Recall & Precision of Query-6. Initial Keyword is “smartphone.”

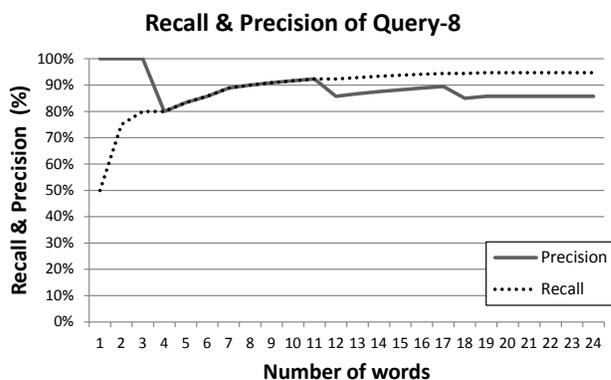


Figure 17. Recall & Precision of Query-8. Initial Keyword is “input.”

VII. CONCLUSION AND FUTURE WORK

We have proposed a complex query navigation system that exploits the search history of social groups. This system recommends candidates for the next search term by calculating the directional relevance along two conceptual dimensions and performing narrowing and sliding operations.

The unique feature of our system is that it combines visualization techniques and semantic correlation computing methods to provide an intuitive user interface dedicated to the modern touchscreens. A social combination function enables the user to utilize the knowledge of social groups to facilitate navigation. We have implemented a prototype system that was able to retrieve and present candidate keywords for multiple queries while reducing the number of touchscreen taps required. Our implementation system is running on the modern slate devices such as iPad and Android.

As future work, we plan to develop a social network-based query recommendation mechanism and evaluate the scalability of complex query navigation in multiple domains. In addition, we are working on developing a context-dependent noise reduction mechanism for words appearing the map because we have recognized that several irrelevant words may appear in the visible area of the word map.

REFERENCES

- [1] Ryo, S. and Shuichi, K., “Cross-Domain Query Navigation System for Touchscreens by Exploiting Social Search History,” in Proceedings of the Seventh International Conference on Internet and Web Applications and Services (ICIW 2012), Stuttgart, Germany, May 27–June 1, 2012, pp. 178-183.
- [2] Cisco, “Global Mobile Data Traffic Forecast Update, 2010-2015” - Cisco Visual Networking Index, February 1, 2011, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf, [retrieved: December 6, 2012]
- [3] Google, “Admod - Tablet Survey,” March 2011. <http://www.ccapitalia.net/descarga/docs/2011-AdMob-TabletSurvey.pdf>, [retrieved: December 6, 2012]
- [4] Harman, D., “Relevance feedback revisited,” in Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1–10, 1992.
- [5] Kelly, D., Gyllstrom, K., and Bailey, E., “A Comparison of Query and Term Suggestion Features for Interactive Searching,” in Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2009), 2009, pp. 371–378.
- [6] Kato, M.P., Sakai, T., and Tanaka, K., “Structured Query Suggestion for Specialization and Parallel Movement: Effect on Search Behaviors,” in Proceedings of the 21st

- International Conference on World Wide Web (WWW2012), ACM, 2012, pp. 389–398.
- [7] Wen, J., Nie, J., and Zhang, H., “Clustering User Queries of a Search Engine,” In Proceedings of the 10th International Conference on World Wide Web (WWW2001), 2001, pp. 162–168.
 - [8] Micarelli, A., Gasparetti, F., Sciarrone, F., and Gauch, S., “Personalized search on the World Wide Web,” in The Adaptive Web, LNCS 4321/2007, 2007, vol. 4321, pp. 195-230.
 - [9] Teevan, J., Dumais, S. T., and Horvitz, E., “Personalizing search via automated analysis of interests and activities,” in Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '05, 2005, pp. 449-456.
 - [10] Gauch, S., Chaffee, J., and Pretschner, A., “Ontology-based personalized search and browsing,” in Web Intelligence and Agent Systems - IOS Press vol. 1, no. 3-4/2003, pp. 219-234.
 - [11] Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., and Boydell, O., “Exploiting query repetition and regularity in an adaptive community-based Web search engine,” User Modeling and User-Adapted Interaction, Apr. 2005, vol. 14, no. 5, pp. 383-423.
 - [12] Dumais, S., Cutrell, E., and Chen. H.. “Optimizing Search by Showing Results in Context,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI2001), 2001, pp. 277–284.
 - [13] Yee, K., Swearingen, K., Li, K., and Hearst, M., “Faceted Metadata for Image Search and Browsing,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI2003), 2003, pp. 401–408.



www.iariajournals.org

International Journal On Advances in Intelligent Systems

✦ ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, ENERGY, COLLA, IMMM, INTELLI, SMART, DATA ANALYTICS

✦ issn: 1942-2679

International Journal On Advances in Internet Technology

✦ ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING, MOBILITY, WEB

✦ issn: 1942-2652

International Journal On Advances in Life Sciences

✦ eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO, SOTICS, GLOBAL HEALTH

✦ issn: 1942-2660

International Journal On Advances in Networks and Services

✦ ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION, VEHICULAR, INNOV

✦ issn: 1942-2644

International Journal On Advances in Security

✦ ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

✦ issn: 1942-2636

International Journal On Advances in Software

✦ ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, IMMM, MOBILITY, VEHICULAR, DATA ANALYTICS

✦ issn: 1942-2628

International Journal On Advances in Systems and Measurements

✦ ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL, INFOCOMP

✦ issn: 1942-261x

International Journal On Advances in Telecommunications

✦ AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA, COCOR, PESARO, INNOV

✦ issn: 1942-2601