ICAS 2015

The Eleventh International Conference on Autonomic and Autonomous Systems

ISBN: 978-1-61208-405-3

May 24 - 29, 2015

Rome, Italy

ICAS 2015 Editors

Pascal Lorenz, University of Haute-Alsace, France

Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance (HLRN), Germany

ICAS 2015

Forward

The Eleventh International Conference on Autonomic and Autonomous Systems (ICAS 2015), held between May 24-29, 2015 in Rome, Italy, was a multi-track event covering related topics on theory and practice on systems automation, autonomous systems and autonomic computing.

The main tracks referred to the general concepts of systems automation, and methodologies and techniques for designing, implementing and deploying autonomous systems. The next tracks developed around design and deployment of context-aware networks, services and applications, and the design and management of self-behavioral networks and services. We also considered monitoring, control, and management of autonomous self-aware and context-aware systems and topics dedicated to specific autonomous entities, namely, satellite systems, nomadic code systems, mobile networks, and robots. It has been recognized that modeling (in all forms this activity is known) is the fundamental for autonomous subsystems, as both managed and management entities must communicate and understand each other. Small-scale and large-scale virtualization and model-driven architecture, as well as management challenges in such architectures are considered. Autonomic features and autonomy requires a fundamental theory behind and solid control mechanisms. These topics gave credit to specific advanced practical and theoretical aspects that allow subsystem to expose complex behavior. We aimed to expose specific advancements on theory and tool in supporting advanced autonomous systems. Domain case studies (policy, mobility, survivability, privacy, etc.) and specific technology (wireless, wireline, optical, e-commerce, banking, etc.) case studies were targeted. A special track on mobile environments was indented to cover examples and aspects from mobile systems, networks, codes, and robotics.

Pervasive services and mobile computing are emerging as the next computing paradigm in which infrastructure and services are seamlessly available anywhere, anytime, and in any format. This move to a mobile and pervasive environment raises new opportunities and demands on the underlying systems. In particular, they need to be adaptive, self-adaptive, and context-aware.

Adaptive and self-management context-aware systems are difficult to create, they must be able to understand context information and dynamically change their behavior at runtime according to the context. Context information can include the user location, his preferences, his activities, the environmental conditions and the availability of computing and communication resources. Dynamic reconfiguration of the context-aware systems can generate inconsistencies as well as integrity problems, and combinatorial explosion of possible variants of these systems with a high degree of variability can introduce great complexity.

Traditionally, user interface design is a knowledge-intensive task complying with specific domains, yet being user friendly. Besides operational requirements, design recommendations refer to standards of the application domain or corporate guidelines.

Commonly, there is a set of general user interface guidelines; the challenge is due to a need for cross-team expertise. Required knowledge differs from one application domain to another, and the core knowledge is subject to constant changes and to individual perception and skills.

Passive approaches allow designers to initiate the search for information in a knowledge-database to make accessible the design information for designers during the design process. Active approaches, e.g., constraints and critics, have been also developed and tested. These mechanisms deliver information (critics) or restrict the design space (constraints) actively, according to the rules and guidelines. Active and passive approaches are usually combined to capture a useful user interface design.

The conference had the following tracks:

- Theory and practice of autonomous systems
- Self-adaptability and self-management of context-aware systems
- Autonomic computing
- System automation
- Cloud computing and virtualization
- Algorithms and theory for control and computation

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the ICAS 2015 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to ICAS 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ICAS 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope ICAS 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of autonomic and autonomous systems. We also hope that Rome, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the historic beauty of the city.

ICAS 2015 Chairs

ICAS Advisory Chairs

Michael Bauer, The University of Western Ontario - London, Canada

ICAS 2015

Committee

ICAS 2015 Advisory Chairs

Michael Bauer, The University of Western Ontario - London, Canada
Radu Calinescu, University of York, UK
Michael Grottke, University of Erlangen-Nuremberg, Germany
Bruno Dillenseger, Orange Labs, France
Mark Balas, Embry-Riddle Aeronautical University, USA
Alex Galis, University College London, UK
Antonio Liotta, Eindhoven University of Technology, The Netherlands
Jacques Malenfant, Université Pierre et Marie Curie, France
Mark Perry, University of New England in Armidale, Australia
Wendy Powley, Queen's University - Kingston, Canada
Nikola Serbedzija, Fraunhofer FOKUS, Germany

ICAS 2015 Technical Program Committee

Jemal H. Abawajy, Deakin University, Australia
Sameh Abdel-Naby, University College Dublin, Ireland
António Abelha, Universidade do Minho - Braga, Portugal
Nouara Achour, USTHB University, Algeria
Carl Adams, University of Portsmouth, UK
Jose Aguilar, Universidad de Los Andes, Venezuela
Javier Alonso, Duke University, USA
Cesar Analide, Universidade do Minho, Portugal
Razvan Andonie, Central Washington University - Ellensburg, USA
Richard Anthony, University of Greenwich, UK
Eva Ibarrola Armendariz, Escuela Técnica Superior de Ingeniería de Bilbao, Spain
Senén Barro, University of Santiago de Compostela, Spain
Ismailcem Budak Arpinar, University of Georgia - Athens, USA
Tsz-Chiu Au, Ulsan National Institute of Science and Technology (UNIST), Korea
Roger Azevedo, McGill University, Canada
Mark Balas, Embry-Riddle Aeronautical University, USA
Michael Bauer, The University of Western Ontario -London, Canada
Matthias Becker, University Hannover, Germany
Janusz Bedkowski, Institute of Mathematical Machines / Warsaw University of Technology, Poland
Julita Bermejo-Alonso, Universidad Politécnica de Madrid, Spain

Michael Grottke, University of Erlangen-Nuremberg, Germany
Jordi Guitart, Universitat Politècnica de Catalunya - Barcelona Tech (UPC), Spain
Odd Erik Gundersen, Verdande Technology, Norway
Mustapha Hamerlain, CDTA, Algeria
Ronny Hartanto, German Research Center for Artificial Intelligence (DFKI GmbH), Germany
Takako Hashimoto, Chiba University of Commerce, Japan
Koen Hindriks, Delft University of Technology, Netherlands
Gerold Hoelzl, Johannes Kepler University, Austria
Wladyslaw Homenda, Warsaw University of Technology, Poland
Wei-Chiang Hong, Oriental Institute of Technology, Taiwan
Marc-Philippe Huget, Polytech Annecy-Chambery-LISTIC | University of Savoie, France
Tim Hussein, University of Duisburg-Essen, Germany
Yoshiro Imai, Kagawa University, Japan
Luis Iribarne, University of Almeria, Spain
Michael Jenkin, York University, Canada
Yiming Ji, University of South Carolina - Beaufort, USA
Richard Jiang, Northumbria University, UK
Yanguo Jing, London Metropolitan University, UK
María José Ibáñez, RIAM I+L Lab (GNOSS), Spain
Fernando Koch, SAMSUNG Research Institute, Brazil
Ah-Lian Kor, Leeds Beckett University, UK
Boris Kovalerchuk, Central Washington University - Ellensburg, USA
Satoshi Kurihara, University of Osaka, Japan
Helge Langseth, NTNU, Norway
Jingpeng Li, The University of Nottingham - Ningbo, China
Fidel Liberal Malaina, University of the Basque Country, Spain
Antonio Liotta, Eindhoven University of Technology, The Netherlands
Hai-Bin Liu, China Aerospace Engineering Consultation Center, China
Angela Locoro, University of Genova, Italy
Noel Lopes, Polytechnic of Guarda, Portugal
Daniela López De Luise, CI2S Labs, Argentina
Hanan Lutfiyya, The University of Western Ontario - London, Canada
José Machado, University of Minho, Portugal
Isabel Machado Alexandre, Iscte-IUL - Instituto Universitário de Lisboa & Instituto de Telecomunicações, Portugal
Prabhat Mahanti, University of New Brunswick, Canada
Sayyed Majid Esmailifar, Sharif University of Technology -Tehran, Iran
Jacques Malenfant, Université Pierre et Marie Curie, France
Elisa Marengo, Università degli Studi di Torino, Italy
Mauricio Marin, Universidad de Santiago and Yahoo! Labs Santiago, Chile
Goreti Marreiros, Polytechnic of Porto, Portugal
Rajat Mehrotra, University of Virginia, USA
Yasser F. O. Mohammad, Assiut University, Egypt / Kyoto University, Japan
Thierry Monteil, LAAS-CNRS, INSA de Toulouse, Toulouse, France

José Moreira, University of Aveiro, Portugal
Masayuki Murata, Osaka University, Japan
Adnan Abou Nabout, University of Wuppertal, Germany
José Neves, Universidade do Minho - Braga, Portugal
Andreas Oberweis, Karlsruhe Institute of Technology (KIT), Germany
Jonice Oliveira, Federal University of Rio de Janeiro, Brazil
Rafael Oliveira Vasconcelos, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil
Michael O'Mahony, University College Dublin, Ireland
Jose Oscar Fajardo, University of the Basque Country, Spain
David Ostrowski, Ford Motor Company / University of Michigan - Dearborn, USA
Maurice Pagnucco, University of New South Wales, Australia
Umberto Panniello, Politecnico di Bari, Italy
Nandan Parameswaran, University of New South Wales - Sydney, Australia
Luis Paulo Reis, University of Minho, Portugal
Loris Penserini, I.I.S. - “L.Donati”, Italy
Mark Perry, University of New England in Armidale, Australia
Steve Phelps, University of Essex, UK
Maria Silvia Pini, University of Padova, Italy
Agostino Poggi, Università degli Studi di Parma, Italy
Wendy Powley, Queen's University - Kingston, Canada
Mariachiara Puviani, DIEF - University of Modena and Reggio Emilia, Italy
Francesco Quaglia, Sapienza Università di Roma, Italy
Kanagasabai Rajaraman, Institute for Infocomm Research, Singapore
Alejandro Ramirez-Serrano, University of Calgary - Alberta, Canada
Martin Randles, Liverpool John Moores University, UK
Marek Reformat, University of Alberta, Canada
Douglas Rodrigues, University of Sao Paulo, Brazil
Paolo Romano, INESC-ID Lisbon, Portugal
Juha Röning, University of Oulu, Finland
Rosaldo Rossetti, University of Porto, Portugal
Lakhdar Sais, Université Lille Nord de France, France
Ricardo Sanz, Universidad Politecnica de Madrid, Spain
Munehiko Sasajima, Osaka University, Japan
Mariano Saura, Polytechnic University of Cartagena, Spain
Christoph Schommer, University Luxemburg, Luxemburg
Paulo Jorge Sequeira Gonçalves, Polytechnic Institute of Castelo Branco, Portugal
Nikola Serbedzija, Fraunhofer FOKUS, Germany
Mohamed Shehab, University of North Carolina at Charlotte, USA
Maxim Shevertalov, Drexel University, USA
Arnab Sinha, INRIA, France
David Šišlák, Czech Technical University in Prague, Czech Republic
Petr Skobelev, Samara State Aerospace University / Smart Solutions, Russia
Flavio Soares Correa da Silva, University of Sao Paulo, Brazil
Nisheeth Srivastava, University of California, USA

Edward Stehle, Drexel University, USA
Claudius Stern, Opsolution GmbH, Germany
Ryszard Tadeusiewicz, AGH University of Science and Technology, Poland
Charalampos Tampitsikas, Università della Svizzera Italiana (University of Lugano), Switzerland
Lijun (Leo) Tang, Microsoft Corporation - Bellevue, USA
Yuqing Tang, Carnegie Mellon University, USA
Patricia Tedesco, Federal University of Pernambuco, Brazil
Ingo J. Timm, Universität Trier, Germany
Manachai Toahchoodee, The University of the Thai Chamber of Commerce, Bangkok, Thailand
Irina Topalova, Technical University of Sofia, Bulgaria
José Manuel Torres, University Fernando Pessoa - Porto, Portugal
Viviane Torres da Silva, Fluminense Federal University, Brazil
Davide Tosi, Università dell'Insubria – Como, Italy
Raquel Trillo Lado, University of Zaragoza, Spain
Egon L. van den Broek, University of Twente, Enschede, Karakter University Center, Radboud University Medical Center Nijmegen, Nijmegen, The Netherlands
Stavros Vassos, Sapienza University of Rome, Italy
Phan Cong Vinh, NTT University, Vietnam
Vladimir Vlassov, KTH Royal Institute of Technology, Sweden
Stefanos Vrochidis, Centre for Research and Technology Hellas - Thermi-Thessaloniki, Greece
Thomas Walsh, Massachusetts Institute of Technology, USA
Bozena Wozna-Szczesniak, Jan Dlugosz University, Institute of Mathematics and Computer Science – Czestochowa, Poland
Nanjian Wu, Chinese Academy of Sciences, China
Reuven Yagel, Ben-Gurion University, Israel
Jinfeng Yi, Michigan State University, USA
Li-Yan Yuan, University of Alberta, Canada
Constantin-Bala Zamfirescu, "Lucian Blaga" University of Sibiu, Romania
Andrzej Zbrzezny, Jan Dlugosz University in Czestochowa, Poland
Chenyi Zhang, Simon Fraser University, Canada
Dieter Zöbel, University Koblenz-Landau, Germany
Albert Zomaya, University of Sydney, Australia

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Support Vector Machine Learning in Multi-Robot Teams 1
Nicol Naidoo and Glen Bright

Intelligent Project Management and Automation Systems 7
Bin Ling, Yanyan Yang, David Ndzi, and Min Gao

Method for Parameter Adjustment for Automated Visual Inspection of Botteled Liquids 15
Irina Topalova

Estimation of Time to Contact from Blurred Images 20
Yukitada Takanashi and Kazuyuki Ito

A Round-Trip Engineering Method for Activity Diagrams and Source Code 24
Keinosuke Matsumoto, Ryo Uenishi, and Naoki Mori

Platform for Autonomous Service Composition 30
Krasimir Baylov, Dessislava Petrova-Antonova, and Aleksandar Dimov

Balancing Centralised Control with Vehicle Autonomy in AGV Systems for Industrial Acceptance 37
Markus Bader, Andreas Richtsfeld, Markus Suchi, George Todoran, Wolfgang Holl, and Markus Vincze

Model-Driven Architecture for Self-Adaptive Context-Aware Message Routing in Pervasive Environments 44
Nachoua Guizani and Jocelyne Fayn

CObAPAS: Combinatorial Optimization based Approach for Autonomic Systems 51
Pedro do Prado, Luis Nakamura, Marcos Santana, Regina Santana, and Omar Cortes

Self-organized Architecture for Sharing Data Streams at Large Scale 57
Nicolas Hidalgo and Erika Rosas

Autonomic Metaheuristic Optimization with Application to Run-Time Software Adaptation 63
John M. Ewing and Daniel A. Menasce

Efficient Management of Cooling Systems in Green Datacenters 73
Ibrahim Safieddine and Noel De Palma

On the Adoption of Multi-Agent Systems for the Development of Industrial Control Networks: A Case Study 77
Hosny Abbas, Mohammed Amin, and Samir Shaheen

Application Design and Profiling of Stream Processing 86

Veronica Gil-Costa, Jair Lobos, and Mauricio Marin

Human Friendly Autonomous Robot using Dempster-Shafer Sensor Fusion and Velocity Potential Field Control 92
Dan Necsulescu, Yu Hu, and Jurek Sasiadek

Enhancing Data Security in Cloud Computing Using a Lightweight Cryptographic Algorithm 98
Sana Belguith, Abderrazak Jemai, and Rabah Attia

Supporting the Neon and VFP Instruction Sets in an LLVM-based Binary Translator 104
Yu-Chuan Guo, Wuu Yang, Jiunn-Yeu Chen, and Jenq-Kuen Lee

Secure Information and Services Management in the Cloud 110
Marek R. Ogiela, Lidia Ogiela, and Urszula Ogiela

Hybrid Autonomous Navigation System Using a Dynamic Fuzzy Cognitive Maps Evolution 114
Marcio Mendonca, Lucia Valeria R. Arruda, Ivan Rossato Chrun, and Esdras Salgado da Silva

Effective Interaction in Asynchronous Multi-Agent Environments for Supply Scheduling in Real-Time 120
Alexander Tsarev, Petr Skobelev, and Dmitry Ochkov

On The Topological Entropy of Continuous-Time Polytopic Systems 126
Graziano Chesi

Task-Space Torque Controller Based on Time-Delay Control 128
Sung-moon Hur, Joonhee Jo, and Yonghwan Oh

Context-Aware Authentication for the Internet of Things 134
Kashif Habib and Wolfgang Leister

Support Vector Machine Learning in Multi-Robot Teams

Nicol Naidoo, Glen Bright

Mechatronics and Robotics Research Group
Department of Mechanical Engineering
University of KwaZulu-Natal
Durban, South Africa
Email: nic.naidoo@gmail.com, brightg@ukzn.ac.za

***Abstract*—In recent years, there has been a great research interest in cooperative mobile robotics. An advancement in industrial technology has seen the need for distributed applications in robotic systems where teams of robots are required to solve tasks intelligently and efficiently. Heterogeneity in robot teams adds complexity to a cooperative system since each member in the team varies in capability which determines its task abilities. The objective of this research paper is to introduce the use of a machine learning system to facilitate cooperation in multi–robot teams. Tests were performed and simulated for mobile robot cooperation in a material handling application during bottleneck conditions.**

Keywords–Multi–robot systems; cooperation; bottleneck; support vector machine; learning.

I. Introduction

Cooperation of Multi–Robot Systems (MRS) have drawn increasing attention in the past two decades since these systems have the ability to perform complex tasks more efficiently compared to single–robot systems [1] [2]. An implementation of a cooperative robot team in a manufacturing environment can, for example, solve the issue of bottlenecks in a production line, whereas the limitations of an individual robot can lead to a lot of problems in terms of time wastage, loss of revenue, poor quality products and dissatisfied customers.

Despite the advantages of MRS, there are still many challenges that exist such as task allocation, collision avoidance, communication, coordinating actions and team reasoning [3]. These challenges together with changing environments and robot heterogeneity, make it impossible for the MRS to predict all of the likely scenarios and thereby act on them. An effective solution to this problem is the incorporation of a learning component to the intelligence of a MRS.

Behaviour-Based Systems (BBS) [4] [5] are learning models that are designed using a bottom–up approach where survival behaviours, such as obstacle avoidance, constitute the low–level robot control and exploration and path planning make up the high–level control component; behaviours are introduced to the model until the desired robot–environment interaction is achieved. Behaviour selection is a key challenge in BBS since it determines which behaviours(s) control the robot at any given time; Reinforcement Learning (RL) has successfully contributed in this regard and has become an area of great interest in the research community [3].

Some other areas of learning mechanisms applied to MRS are artificial neural networks [6] and genetic algorithms [7] [8]. The focus of this paper is to discuss the use of the Support Vector Machine (SVM) learning algorithm in MRS. SVM learning is a supervised, classification or inductive learning scheme where the computing system learns from the database of past experiences to predict future outcomes; it has been successfully implemented in many applications, such as bioinformatics, and text and image recognition.

This paper aims to broaden its use in MRS applications where cooperation among robot team members is a key requirement. The remainder of the paper is structured as follows: Section II discusses the background and theory of SVMs, in particular, linear and non–linear classifiers, and some popular SVM libraries that can be used in applications; Section III discusses the design, implementation, and test results of the SVM learning system in a material handling application; Section IV concludes the paper and introduces further work to the research.

II. SVM background

SVM learning is related to statistical theory [9] and was first introduced as a classification method in 1992 [10]. It is widely used in bioinformatics due to its accuracy and ability to work with high–dimensional space data. The standard SVM is a binary linear classifier (commonly referred to as the linear SVM) which predicts whether an input belongs to one of two possible classes; this is accomplished by first building a model from a set of training examples, each consisting of input data that are mapped to the corresponding class label. SVM non–linear classifiers can be created by using non–linear kernel functions, further discussed in Section II-B.

A. SVM linear classifiers

In order to gain an intuition on what support vectors actually are and how they are used to create learning models a few preliminary mathematical terms will now be introduced. Given some training data set, *D*, with *n* points:

$$D = \{(\mathbf{x}_i, \mathbf{y}_i), \mathbf{x}_i \in R_m, \mathbf{y}_i \in \{-1, 1\}\}_{i=1}^{n} \qquad (1)$$

The boldface **x** term is a vector with training example inputs $\mathbf{x}_i$; each $\mathbf{x}_i$ has an *m*–dimensional size of *m* features. The classifier term, $\mathbf{y}_i$, is either -1 or 1 and indicates the class to which each point $\mathbf{x}_i$ belongs.

In Figure 1 (a), the training examples are classified into positive and negative classes. The hyperplane, *H*, is the *decision boundary* that divides the regions between positive and negative classes. The decision boundary is said to be *linear* since the examples are linearly separable and a classifier with

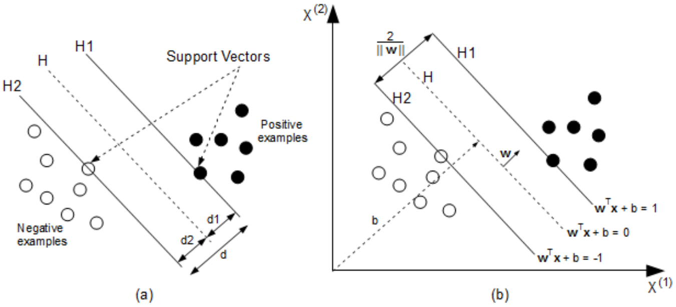


Figure 1. (a) Hyperplanes and margins. (b) Margin classifiers

a linear decision boundary is called a *linear classifier*. *H1* and *H2* are lines that intersect the *support vectors*, these are the training examples that are closest to the decision boundary and they determine the margin (*d1* and *d2*) at which the two classes are separated from the hyperplane (or decision boundary). The SVM algorithm is also termed as the *large margin classifier* since its goal is to maximise the margin *d* for a set of classified training examples.

Figure 1 (b) is an extension to (a) and shows the training examples on a two dimensional feature space with features $x^{(1)}$ and $x^{(2)}$. A linear classifier is based on a linear function of the form:

$$f(x) = \mathbf{w}^T\mathbf{x} + b \tag{2}$$

where **w** is commonly known as the weight vector and *b* is the bias. The product between **w** and **x** is known in linear algebra as the dot product and is defined as $\mathbf{w}^T\mathbf{x} = \sum_i w_i x_i$. The equation for the hyperplane is:

$$H : \mathbf{w}^T\mathbf{x} + b = 0 \tag{3}$$

where the purpose of the bias can be seen as moving the plane away from the origin, i.e., if *b=0* the hyperplane would go through the origin. Equations (4) and (5) are related to planes *H1* and *H2*:

$$H1 : \mathbf{w}^T\mathbf{x} + b = 1 \tag{4}$$

$$H2 : \mathbf{w}^T\mathbf{x} + b = -1 \tag{5}$$

and are equated to 1 and -1 respectively due to the definition of the classifier term, $\mathbf{y}_i$ in (1). Using geometry and referring to Figure 1 (b), the margin between *H* and *H1* is 1/$\|\mathbf{w}\|$, where $\|\mathbf{w}\|$ is the length of the vector **w** and is given by $\sqrt{\mathbf{w}^T\mathbf{w}}$; hence the margin between *H1* and *H2* is 2/$\|\mathbf{w}\|$. In order to maximise the margin, $\|\mathbf{w}\|$ must be minimised subject to the following constraints which are added to prevent data points falling into the margin:

$$\mathbf{w}^T\mathbf{x}_i + b \geq 1, \quad \{for\ \mathbf{y}_i = 1\} \tag{6}$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1, \quad \{for\ \mathbf{y}_i = -1\} \tag{7}$$

Equations (6) and (7) can be combined to form:

$$\mathbf{y}_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad \{for\ 1 \leq i \leq n\} \tag{8}$$

Minimising $\|\mathbf{w}\|$ subject to (8) is a *constrained optimisation problem* and solving it requires using the method of *Lagrange multipliers*. A method that can be used to obtain a dual formulation, expressed in terms of α_i variables [11]:

$$\text{maximise } \alpha\text{:} \ \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T\mathbf{x}_j \tag{9}$$

$$\text{subject to: } \sum_{i=1}^{n} y_i\alpha_i = 0, \quad \alpha_i \geq 0 \tag{10}$$

The dual formulation also defines the weight vector in terms of the training examples:

$$\mathbf{w} = \sum_{i=1}^{n} y_i\alpha_i\mathbf{x}_i \tag{11}$$

B. SVM non–linear classifiers

In most SVM classification problems, the data set is not linearly separable. Literature [10] solves this challenge by mapping the original finite dimensional space into a higher dimensional space making the separation much easier in that space, as illustrated in Figure 2.

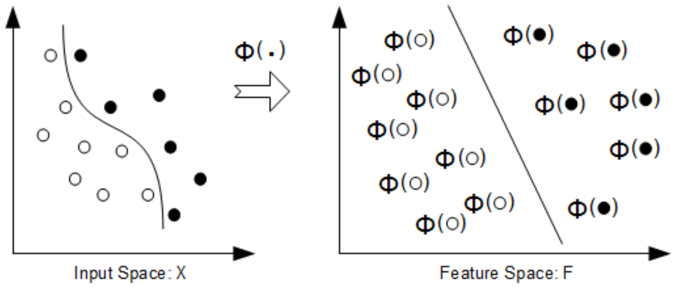


Figure 2. Non-linear classification mapping

The mapping is achieved by the use of *Kernel functions* and the dot product property in the linear SVM algorithm. The $\mathbf{x}_i^T\mathbf{x}_j$ terms in (9) are replaced by the kernel function, *K*:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) \tag{12}$$

which can represent (among others) a *polynomial*, *gaussian*, or *hyperbolic function* [12]. The linear classifier is also known as the *linear kernel*.

C. Multi–class SVM

SVMs are inherently binary classifiers however, there are many applications where multiple classifications are required. The common method of solving the *M-class* problem is to divide it into multiple binary classification problems [13]:

- One-vs-All: This method constructs *N* binary SVM classifiers, where *N* represents the number of classes. Every *i*-th SVM is trained to differentiate the training examples of the *i*-th class from the examples of the other classes. At the classification phase, samples are classified in accordance to the highest output function among all the SVMs.
- One-vs-One: This strategy constructs one SVM for every pair of classes, hence for an *M-class* problem of *N* classes, *N*(*N*-1)/2 SVMs are trained. A maximum-wins voting concept is used where each SVM classifier assigns the sample to one of the two classes and

the number of votes for the assigned class increases by one; in the end, the class with the most votes determines the classification of the sample.

Another approach to the *M-class* problem, which avoids the use of multiple binary classification problems, involves the application of a single optimisation model [14].

D. SVM software libraries

Over the past two decades there has been a wide interest in SVM algorithms which has led to the development of many solvers for SVM optimisation problems. Two popular open source solvers are LIBSVM [15] and SVM^{light} [16]. These solvers form excellent tools for researchers since they eliminate the vast quantity of time that could be spent on the complex software development of SVM optimisation algorithms and thus allow the scientist to focus on the primary components of the research. The LIBSVM library was used in this research.

III. Multi-robot cooperation application

The multi–robot cooperation research was tasked for advanced manufacturing environment applications where dissimilar (or heterogeneous) mobile robots are used in discrete processes. The idea of cooperation between robots when there is a need can prevent bottlenecks, improve material flow and thus contribute to the upkeep of a good supply chain management system. The objective of the research is to aid any member in a team of heterogeneous robots in task decision making. Each robot in the system must be capable of moving autonomously in the known environment while avoiding obstacles and maintaining a teamwork approach in the resolution of common goals. An essential component of the design is the machine learning algorithm which is used to predict suitable goal destinations for each mobile robot, given a set of input parameters.

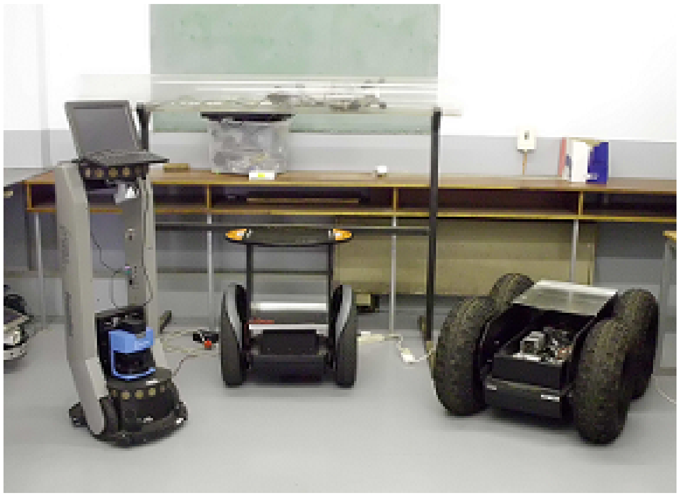


Figure 3. Mobile robot hardware used for the research

The three mobile robots (Figure 3) used in the research were the Performance PeopleBot, the Segway RMP200, and the Segway RMP400. The platforms were chosen on the basis of their availability; they are mainly used for research purposes and not suited for manufacturing environment applications, which is acceptable for the research since the objective is to establish the concept of a cooperating team of heterogeneous mobile robots, irrespective of their abilities and functionality.

A. Material handling application description

The objectives of this research were tested in a material handling application, as illustrated by the Supervisory Control and Data Acquisition (SCADA) screenshot shown in Figure 4. The application shows a resource buffer (“R”), a storage buffer (“S”), 6 process buffers (“B1”–“B6”), 3 machines (“M1”–“M3”), and a conveyor; it was designed in this manner to demonstrate the cooperative ability of the system during bottleneck and fault conditions. The application was set up for the PeopleBot to transport material from “R” to “B1”, the RMP200 move material from “B4” to “B5”, and the RMP400 to finally move the end product from “B6” to “S”.

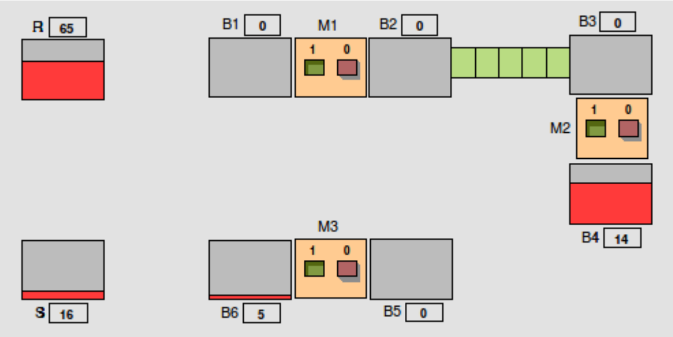


Figure 4. Material handling application for the research

The numbers within the blocks shown in Figure 4 represent the quantity of material in the buffer and the buffer levels are illustrated as a percentage of their total capacity, thus the bottlenecks in the process can be seen at a glance during production. The calculations for the quantity of material, buffer capacities, and machine process rates are all done in the simulation program which is located in the SCADA component of the system, the details of which are beyond the scope of this paper.

During the implementation and debug phase of this research, bottleneck conditions were intentionally created by altering: 1) the material handling capacities of the robots, 2) the machine efficiencies, and 3) the buffer capacities.

B. Design overview

Figure 5 shows the design overview of the Mechatronic system. The scope of the design consists of an integration of the following components:

- Robot hardware
- Middleware
- Agent program
- SCADA

The *robot hardware* comprises of the mechanical robot (PeopleBot, RMP200, RMP400), the sensors (LRF, sonars), and the actuators (drives, motors). The *middleware* layer is necessary since it is responsible for interpreting the high level (agent program) commands and presents them to the sensors and actuators through the use of low level software driver modules.

ISBN: 978-1-61208-405-3

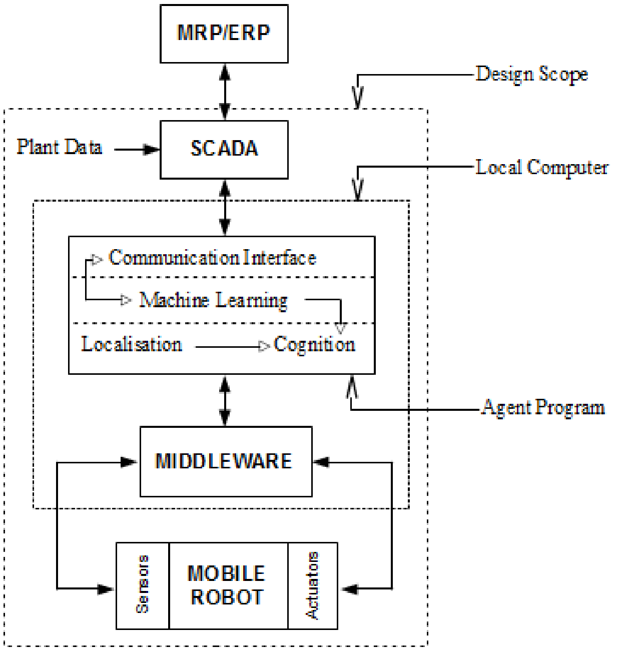


Figure 5. Design overview of the Mechatronic system

The *agent program* is the robot's decision making component in the system design as it determines which task (primary or secondary) is required by the robot at a specific point in time. In addition to the *localisation* and *cognition* modules, the agent program contains the *machine learning* module which was incorporated in the system design due to the following benefits:

- Robot heterogeneity and task taxonomy: due to the different capabilities of each robot together with the variations in tasks, the system is required to identify whether or not a particular robot can perform a secondary task when required. An integrated learning system will ensure that each robot goes through an engineering teaching process so that the robot "agent" can identify itself as a helping agent when the need (bottleneck) arises.

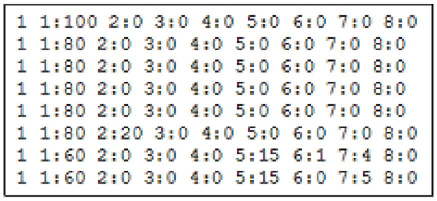
- Manufacturing environment reconfiguration: changes in the environment, caused by the manufacturing of different products or the implementation of new machinery, will have a minimal impact on the cooperative function of each robot since the learning module ensures that robot agents are re-taught accordingly. A further advantage is the saving of money and resources that would have been required to reconfigure the robots to adapt to the new environment.

The *agent program* also comprises a *communication interface* which sends, receives and processes data packets to/from the plant SCADA system. The SCADA is a vital component in the manufacturing plant automation system since it makes process information available to operators and engineers for the purpose of monitoring and control.

C. SVM implementation

The LIBSVM library was used in the agent program for the train and prediction algorithms, and the polynomial kernel was chosen as the non–linear SVM kernel function. There are two phases to the SVM algorithm:

- the *learning* phase, where agents are taught by the system on the best goal location to follow. The teaching process can take place in an offline (simulation) environment, or online through the Graphical User Interface (GUI) interface of the SCADA system.
  The objective of the learning phase is to build a knowledge database of SVM features with training examples. Figure 6 is an extract of the "train.txt" file that contains the training examples. The SVM features in the file (labeled 1 to 8) are the buffers in the manufacturing application and the training examples (the values positioned to the right of the colons) are the number of materials in each buffer. The (output) goal location for the robot is the first number in each line of the file.



```
1 1:100 2:0 3:0 4:0 5:0 6:0 7:0 8:0
1 1:80 2:0 3:0 4:0 5:0 6:0 7:0 8:0
1 1:80 2:0 3:0 4:0 5:0 6:0 7:0 8:0
1 1:80 2:0 3:0 4:0 5:0 6:0 7:0 8:0
1 1:80 2:0 3:0 4:0 5:0 6:0 7:0 8:0
1 1:80 2:20 3:0 4:0 5:0 6:0 7:0 8:0
1 1:60 2:0 3:0 4:0 5:15 6:1 7:4 8:0
1 1:60 2:0 3:0 4:0 5:15 6:0 7:5 8:0
```

Figure 6. Train.txt file extract with SVM features and training examples

- the *train–prediction* phase uses the data collated in the learning phase (i.e., the data contained in the train.txt file) to generate training models for each agent; the goal output for each agent is then accomplished by using the current data values (obtained from the data packet) as inputs to the prediction algorithm. The current data values represent the immediate status of the manufacturing process; they are stored as a string of data in the "test.txt" file which is used as an input to the SVM prediction algorithm. Figure 7 illustrates the entire process of training, building the model, and predicting the goal output for each robot in the system.

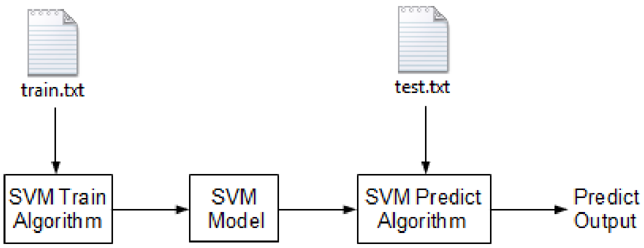


Figure 7. Process of the SVM train–predict phase

D. Simulation results and discussion

This section produces the results of the tests performed during the simulation of the system. Bottlenecks were created by varying the load carrying capacities of the robots, however, there were other options by which this could have been done, namely: 1) vary the machine or conveyor efficiencies, and 2) change the buffer capacities.

During the SVM teach phase of the tests, the PeopleBot was taught to help the RMP200 at the bottleneck. Figure 4 showed a screenshot of the material handling application, where the PeopleBot's primary task is to move materials from the resource buffer ("R" or "B0") to "B1", the RMP200 has the single task of transporting material from "B4" to "B5", and the RMP400 also has a single task of moving the final product from "B6" to the storage buffer ("S" or "B7").

A bottleneck was created at "B4" by reducing the load carrying capacity of the RMP200 from 20 materials to 5 materials. The capacity of the PeopleBot remained the same (at 20 materials), this ensured that the material build up rate at B4 was greater than the buffer process rate, resulting in a bottleneck.

Four types of simulation tests were performed:

- normal operation: the load carrying capacities of the robots were configured to prevent bottleneck conditions.
- bottleneck condition: the load carrying capacities of the robots were configured to promote bottleneck conditions.
- cooperation at the bottleneck: a robot agent was allowed to help another agent at the bottleneck.
- cooperation during a robot fault: a robot agent was allowed to take over the tasks of the faulty robot so that the possibility of the occurrence of a bottleneck is reduced.

A discussion of all four types of tests is beyond the scope of this paper, hence only the *bottleneck condition* and *cooperation at the bottleneck* cases will be discussed.

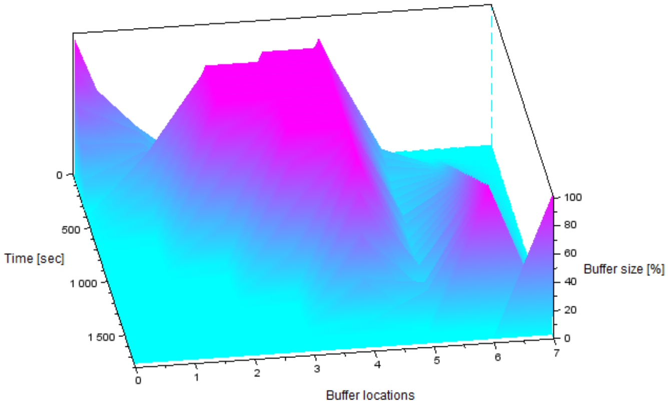


Figure 8. Material distribution graph: bottleneck condition

The material distribution graph for the *bottleneck condition* simulation test is given in Figure 8. The graph has three axes: the x–axis represents the buffer locations, ranging from 0 (buffer B0) to 7 (buffer B7); the y–axis represents the time (in seconds) of the simulation; the z–axis gives the number of materials, in a percentage, at each buffer location. The percentage is calculated by the following equation:

$$B_{size} = \frac{B_{num}}{B_{cap}} * 100 \tag{13}$$

where B_{num} is the number of materials in the buffer and B_{cap} is a constant which represents the number of materials that the buffer can contain, i.e., the buffer capacity.

The visual trend in the graph shows a decrease in material count at the resource buffer (which was initialised with 100 materials) and an increase in material count at the storage buffer, towards the end of the simulation. Table I gives more detail to the *bottleneck condition* simulation and lists the values of some test parameters such as the total simulation (or production) time and the total operation time of each robot agent.

TABLE I. SIMULATION RESULTS FOR THE BOTTLENECK CONDITION

Test parameter	*Value*
Total simulation time	1763 sec
Agent 1 load capacity	20 materials
Agent 2 load capacity	5 materials
Agent 3 load capacity	100 materials
Agent 1 operation time	349 sec (19.8%)
Agent 2 operation time	1520 sec (86.2%)
Agent 3 operation time	93 sec (5.3%)
Buffer 2 @100%	282 sec (16.0%)
Buffer 3 @100%	594 sec (33.7%)
Buffer 4 @100%	936 sec (53.1%)

Agents 1, 2 and 3 are the PeopleBot, RMP200 and RMP400 respectively. The values within brackets in the table are the percentages of the total simulation time. The large simulation time for the *bottleneck condition* is due to the bottleneck at buffer 4, where the RMP200 cannot transport the required amount of material to keep up with the incoming rate at the buffer. The bottleneck problem caused a cascaded effect (depicted in Figure 8) to fill up buffer 3 and buffer 2. The purpose of the *bottleneck condition* simulation was two–fold: 1) to emphasise the impact of the bottleneck on the production system, and 2) to set the stage for an implementation of the cooperative learning system in mitigating the bottleneck.

The *cooperation at the bottleneck* simulation was performed by allowing the SVM–trained PeopleBot agent to assist the RMP200 agent at the bottleneck (buffer 4), hence the PeopleBot executes its primary task of transporting material from B0 to B1 as well as "cooperates" by effecting its secondary task of moving material from B4 to B5. The material distribution graph in Figure 9 reflect the results of the cooperative learning system.

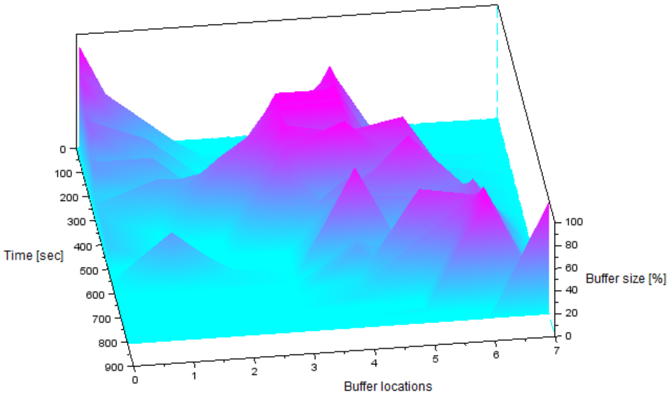


Figure 9. Material distribution graph: robot cooperation at bottleneck

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-405-3

An analysis of the SVM output results in the subplot of Figure 10 gives an interesting perspective on the periods at which the algorithm determines the assistance of the PeopleBot at the bottleneck. The SVM outputs for the PeopleBot agent are either "1" or "2", representing the primary or secondary task respectively. During the teach phase, the PeopleBot agent was taught to assist at B4 when the size of B0 is low and when the sizes of B4 and/or B3 are high. The effect of the teaching exercise is clearly shown in Figure 10 since the SVM predictions are "2" during conditions where the test parameters of the SVM features (i.e., the buffer sizes) are approximately the same as the SVM training examples.

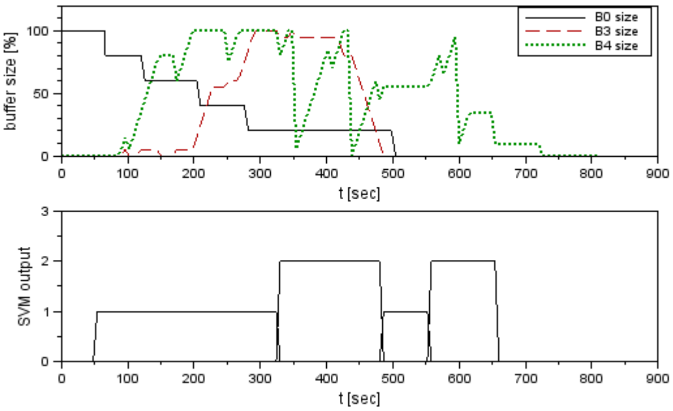


Figure 10. PeopleBot SVM outputs: cooperation at bottleneck

TABLE II. SIMULATION RESULTS FOR THE COOPERATION AT THE BOTTLENECK CONDITION

Test parameter	*Value*
Total simulation time	809 sec
Agent 1 load capacity	20 materials
Agent 2 load capacity	5 materials
Agent 3 load capacity	100 materials
Agent 1 operation time	600 sec (74.2%)
Agent 1 primary task	62.5%
Agent 1 secondary task	37.5%
Agent 2 operation time	678 sec (83.8%)
Agent 3 operation time	91 sec (11.3%)
Buffer 3 @100%	42 sec (5.2%)
Buffer 4 @100%	138 sec (17.1%)

Table II lists the total simulation time of 809 seconds—a 54% reduction in comparison to the previous simulation case. The table also reflects the task distribution percentage for agent 1: the SVM algorithm determined the secondary goal for the PeopleBot 3 times out of a total of 8 iterations in the simulation, i.e., the PeopleBot spent 37.5% of its operation time on the secondary task and 67.5% on its primary task. The simulation also resulted in an elimination of buffer 2 from the bottleneck cascade and showed reduced buffer–full times of buffer 3 and buffer 4 to 5.2% and 17.1%, respectively.

IV. CONCLUSION

The main objective of the research was the demonstration of a cooperative robot system using a machine learning approach. This objective was achieved by the successful performance of the SVM algorithm, where the bottlenecks were alleviated by the cooperating agent, significantly improving the manufacturing production times. The SVM learning algorithm essentially predicts and determines the goal tasks of each robot agent in the network by using a database of training examples.

The research discussed in this paper broadens the use of SVM algorithms (and potentially other supervised learning algorithms) in the area of multi–robot systems and manufacturing applications. The attraction of a learning based system is the semi–elimination of hard coded programmed solutions for specific scenarios; the learning system can adapt to dynamic environments and plant reconfiguration conditions.

Further work to this research will see the implementation of a reinforced learning system where the agents dynamically learn the "positive" and "negative" examples from the environment without going through a training exercise facilitated by the robot operator. Another desired modification to the system is the use of an automated selection of a training database in a suite of databases, this is useful when an agent has to solve a variety of problems, requiring the employment of multiple sets of training data.

REFERENCES

[1] Y. Cao, A. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," Autonomous Robots, vol. 4, no. 1, Kluwer Academic Publishers Hingham, MA, USA, 1997, pp. 7–27.

[2] L. E. Parker, T. Arai, and E. Pagello, "Advances in Multi-Robot Systems," IEEE Transactions on Robotics and Automation, vol. 18, 2002, pp. 655–661.

[3] E. Yang and D. Gu, "Multiagent reinforcement learning for multi-robot systems: A survey," In Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games (CIG05), Essex, UK, 2005.

[4] L. E. Parker, "Alliance: An architecture for fault tolerant multi robot cooperation," IEEE Transactions on Robotics and Automation, vol. 14, no. 2, IEEE, 1998, pp. 220–240.

[5] M. J. Mataric, "Learning in behaviour-based multi-robot systems: policies, models and other agents," Journal of Cognitive Systems Research, vol. 2, 2001, pp. 81–93.

[6] S. Bhattacharya and S. Talapatra, "Robot motion planning using neural networks: A modified approach," International Journal of Lateral Computing, vol. 2, no. 1, World Federation on Lateral Computing, 2005, pp. 9–13.

[7] E. Sahin and W. Spears, "Swarm robotics, a state of the art survey," Lecture notes in Computer Science, vol. 3342, 2005.

[8] N. Naidoo, G. Bright, and R. Stopforth, "Material flow optimisation in flexible manufacturing systems," In Proceedings of the 6th IEEE Robotics and Mechatronics Conference (RobMech), Durban, South Africa, 2013, pp. 1–5.

[9] V. Vapnik, The Nature of Statistical Learning Theory. Springer–Verlag, New York, 2000.

[10] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," In Proceedings of the 5th annual workshop on Computational learning theory, ACM, New York, USA, 1992, pp. 144–152.

[11] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning, vol. 20, no. 3, Kluwer Academic Publishers, Boston, 1995, pp. 273–297.

[12] B. Schlkopf and A. Smola, Learning with Kernels. MIT Press, 2002.

[13] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," IEEE Transactions on Neural Networks, IEEE, 2002.

[14] K. Crammer and Y. Singer, "Algorithmic implementation of multiclass kernel-based vector machines," Journal of Machine Learning Research, vol. 2, no. 1, ACM, 2002, pp. 265–292.

[15] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," retrieved: March 2015. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/papers/libsvm.pdf

[16] T. Joachims, Making large-scale support vector machine learning practical: Advances in Kernel Methods. MIT Press, 1998.

Intelligent Project Management and Automation Systems

Bin Ling, Yanyan Yang, David Ndzi
School of Engineering
University of Portsmouth
Portsmouth, UK
email: bin.ling@myport.ac.uk, {linda.yang;
david.ndzi}@port.ac.uk

Min Gao
School of Software Engineering
Chongqing University
Chongqing, China
email: gaomin@cqu.edu.cn

***Abstract*— Effective management of projects is becoming increasingly important for any type of organisation to remain competitive in today's dynamic business environment due to pressure of globalisation. Planning a project with proper considerations of all necessary factors and managing a project to ensure its successful implementation are facing a lot challenges. Initial stage in planning a project is costly, time consuming and usually with poor accuracy on cost and effort predictions. On the other hand, detailed information for previous projects may be buried in piles of archived documents, which make it increasingly difficult to learn from the previous experiences. Project portfolio has been brought into this field aiming to improve the information sharing and management among different projects. However, the amount of information that could be shared is still limited to generic information. In this paper, we design and implement a novel software system iPAS, which automatically generate a project plan with effort estimation of time and cost based on data collected from previous completed projects in standardised industries. To maximise the data sharing and management among different projects, a method of using product-based planning from PRINCE2 methodology is proposed. iPAS has been trialed with cases in two organisations, which clearly shows the business benefits of autonomic project management. It reduced effort to plan new projects and manage project portfolio and decreased estimation bias thereby reducing operational risk. It also automatically benchmarked performance against company best practices.**

Keywords-autonomatic project management; product-based planning; best practice; PRINCE2.

I. INTRODUCTION

In recent years, many engineering companies have spent a great deal of time bidding for Whole Life Cycle (WLC) projects from clients [1]. Most of the project planning and associated cost are developed almost from scratch, even when elements of projects are similar to those bid for in the past. Since the bidding proposal must be built around a sound and well-thought-out estimated project plan, which addresses the cost, time spent and quality to generate the final product to be responsive to clients' delivery requirements, it will take considerable time and therefore incurs resource costs, which could be a big cost saving.

Frequently, the best practice of assessing through life support resources in the engineering services sector is to benchmark against a similar and previous project [3] by using historical data. Best practice is defined as the most efficient (least amount of effort) and effective (best results) way of accomplishing a task or a deliverable, based on repeatable procedures that have proven themselves over time for large numbers of people [2]. Benchmarking is considered as a technique to provide a systematic approach to improving business production efficiency and profitability through comparing and analysing the values from varying resources. Thus, benchmarking and utilising best project practice are the key issues for enterprises to persist in contract competition and project planning.

Currently, most best project practices are made explicit in terms of persistent data from operational processes or activities, but underlying influencing factors remain implicit. The risk of such practice is the cost estimation will not take into account other factors, such as different environment, technology advances and different customer profiles [3].

On the other hand, main stream project management methods nowadays are process or activity based. Therefore, the granularity of information is collected merely at the activity level. Project portfolios which assist the decision makers on corporate strategy and project management practices are also mainly represented based on process. Project information sharing happens only at the activity level, or at the project level, in this case. At activity level, information is not easily sharable due to the fact that new technologies, process re-engineering and different personnel preferences may all affect the practices of conducting project activities. The vast amount of information in between which contains the best practices of working on certain products (deliverables) is not even collected. There is an emerging requirement from industries to have a tool to use good practices or lessons learned from previous projects to guide the new projects.

This paper introduces a web based adaptive project information sharing and management system - iPAS (Intelligent Project Automation Systems). iPAS is developed by following modern software engineering methodologies, PRINCE2 [4] principles and the practical experience of project managers. iPAS consists of four main project management functions: project planning, progress monitoring, project reports and human resource allocation. It was designed for managing engineering projects, but its principles could also be applied to other project disciplines.

The rest of the paper is organized as follows. Section 2 introduces the state of the art of current project management systems. Section 3 presents the overview of the iPAS system and related PRINCE2 techniques, followed by introducing the major functions of iPAS in Section 4. Section 5 provides evaluation results of the system. The final section concludes the whole paper.

II. Project Management Systems in the Market

A number of commercial tools have been created for project information sharing and project management. These commercial applications have been adopted by industry at a remarkable rate. For example, Microsoft Project [16] is able to develop project plans with Gantt charts [17], assigning resources to tasks and tracking progress; MindManager [5] can easily convert brainstorm maps into process diagrams, create standard templates so every project has continuity and can easily be exported to the Microsoft Office suite; @TASK [6] has features such as interactive Gantt charts, calendar views and project group lists that are designed to minimise downtime and make data management easy; ASTA Power Project [7] is a standalone software to do the time planning, project progress monitoring and resource management; Instant Business Network (IBN) Project Management [8] provides a cost-effective and flexible approach to repeating success and re-using a unified system to consolidate corporate information into a single web portal.

However, the most widely used project management features of these applications are fairly conventional. For instance, the classical feature of graphical plan and critical path analysis, display the Gantt chart view by default encourages users to focus on task or activity scheduling too early, rather than identifying objectives and deliverables. Moreover, plans generated by these applications are based on activities, which make it difficult to perform the benchmark because different project users may have different approaches to deliver the same product. The detailed information collected at the activity level can be useful for future project planning only when the same work practices are followed. In addition, due to no shared central database to store historical data, these project management applications cannot do benchmarking from previous projects and use the historical data to produce an automated project plan.

III. iPAS System

As mentioned previously, many engineering companies spent considerable time bidding for projects by developing project plans from scratch. It also means that bids are not always consistent and sometimes contain inaccuracies, which can be costly if the project is won and the cost profile is proved to be wrong. Furthermore, on contract award it is difficult to substantiate existing data on project success to improve customer confidence.

Although benchmarking has been brought into project management, the risk is the effort estimation will not take into account other factors, such a different environment, technology advances and different user profiles. Companies, such as Dytecna [1] have previously been financially penalised by poor benchmarking techniques. Dytecna is an engineering company providing engineering service solutions for governments and commercial customers, both in the United Kingdom and overseas. Its core business activities include Systems Engineering, Whole Life Support, Manufacturing, and Asset Management/Health Monitoring Systems. A research project was proposed by Dytecna a couple of years ago to employ benchmarking techniques to improve current information sharing and management of the whole life cycle of projects based on the best practices from historical data, and find a method to analyse the completed and existing projects to convert Activity Based project information into Product Based information. The outcome of the project is an adaptive project information sharing and management system – iPAS which is based upon the principles of the best practices and the methodologies from PRINCE2 to manage the whole lifecycle of project.

A. *iPAS Overview*

The philosophy behind the design of iPAS is to facilitate system learning from previous projects in light of benchmarking criteria and present to the project manager a manageable amount of easily-derived information organised to give insight, information, or alerts about project status.

To achieve this goal, iPAS is designed to intelligently assist with the through life management of projects based on best practice and experience from previous project profiles. The system is expected to automatically deliver project plans to match customer requirements and provides a mechanism for continuous monitoring of project execution via benchmarking and generation of project reports.

In order to fully utilise this service, an additional consultancy service package is provided to help customers to break down their products into sub-products or work packages in accordance with PRINCE2 project management principles such as Product Based Structure and Product Flow Diagram. This data is stored in a central database for analysis and benchmarking enabling project managers to control their projects with greater precision. Thus, iPAS system consists of two main parts: a web based project management tool which allows users to access it anytime and anywhere, and a consultancy service (shown in Figure 1).

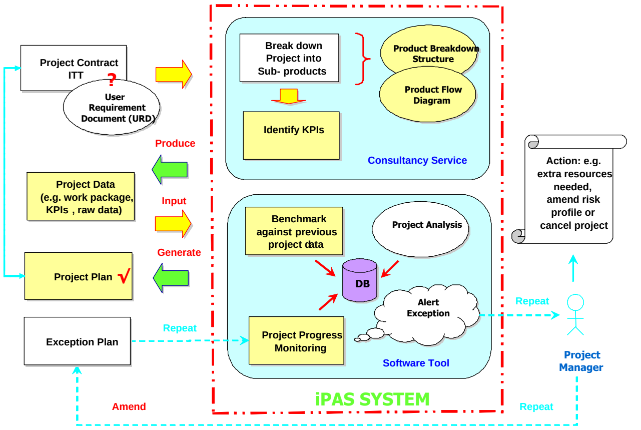

Project Contract
ITT
?
User Requirement Document (URD)
Produce
Project Data (e.g. work package, KPIs , raw data)
Input
Generate
Project Plan √
Exception Plan
Repeat
Amend
Break down Project into Sub- products
Product Breakdown Structure
Product Flow Diagram
Identify KPIs
Consultancy Service
Benchmark against previous project data
Project Analysis
DB
Alert Exception
Project Progress Monitoring
Software Tool
iPAS SYSTEM
Action: e.g. extra resources needed, amend risk profile or cancel project
Repeat
Project Manager
Repeat

Figure 1. iPAS Overview.

When a company receives Invitation To Tender (ITT) to submit a project bidding, the user requirement document will be used as an input of the consultancy service part of the system. With the help from project management domain experts, the project will be broken into sub-products by using Product Breakdown Structure (PBS) [4] and Product Flow Diagram (PFD) [4] techniques, relevant project data such as work packages and identified Key Performance Indicators (KPIs) will be produced as an output of the consultancy service part.

These raw project data then will be stored into iPAS software too. This tool will automatically analyse and provide forecasts for newly entered ITT data, by benchmarking it with historical project data. The tool is able to produce accurate plans through analysis of entered criteria against benchmarked data, which helps project managers to decide the project plan. If the bidding is successful and when the new project starts, the tool will also monitor the project plan by measuring identified key performance indicators (e.g., time, cost and quality) in produced work packages during the project life cycle. The project management team will receive alerts if any project activity goes wrong or beyond the controls during the project progress. Immediate action can be taken, such as allocate extra resources when needed or amend the risk profile, or even cancel the project, to ensure the project status is healthy. A final report can then be generated to summarise the project. Benchmarked data will be updated in a central database at the end of the project to improve the analysis provided to subsequent projects or biddings.

B. *Product Based Planning Technique*

Project planning is about effort estimation including time, cost and resources, which is based on expert judgment and analogy using historical data from completed projects. Consistency in historical data gathering is the key to reliable estimates [9][10]. Especially data for status report in business project management system should be collected during and after project, but in rare cases automatic data capture may be available [11].

iPAS uses product breakdown structure to delineate the project scope and define a list of deliverable products to be constructed during the project. The products must be identified before the activities are defined since the object of the project is to produce deliverables. As mentioned in last section, in accordance with PRINCE2 project management principles, users need to break down projects into work package sized products (includes intermediate documentary products and final end-products) before fully utilising iPAS. Therefore, it is expected that products (or work packages) are identified through PBS and PFD before using iPAS tool.

Compared to activity-based planning or process-based planning, a significant advantage of product based planning is to do with reporting. It can more precisely control the scope of the project and focus only on what is really needed to meet the business case. Products are either finished or not, activities can be 95% finished for a long time even though work is taking place [12]. One tends to forget things that have to be done to complete a project. This method captures them all, reducing the chance that any will be overlooked. Another significant advantage of it is that it will be much easier to benchmark with same or similar products because different project users may have different processes or approaches to delivering the same product, but the properties (e.g., quality, cost and time) used to measure the completed product should be the same.

Here is an example of project plan to integrate current IT operations into a “Web Based Information Management System (WBIMS)” in Dytecna Ltd. In PRINCE2, the top level of products is known as “project products”. For WBIMS project, these are subdivided into three main categories, as shown in Figure 2 represented with diamond shape.

Management products are those products associated with the planning and control of the project. They include Project Initiation Documents (PID), project plan, checkpoint reports and so on. Quality products are separated from Management products, they are associated with the definition and control of quality, quality plan, product descriptions, quality review reports, and project issues report. Specialist products are those things that the project has been setup to create. It can be broken down into other three sub categories [4]:

- Analysis Products.
- Development Products and
- Implementation Products

Each category respectively includes a few products underneath. For example, Website is a deliverable of Development Products Group during the system design and development stage, while the Tested system and Implemented system (signed off acceptance system) are the deliverables of the Implementation Products Group in the implementation stage.

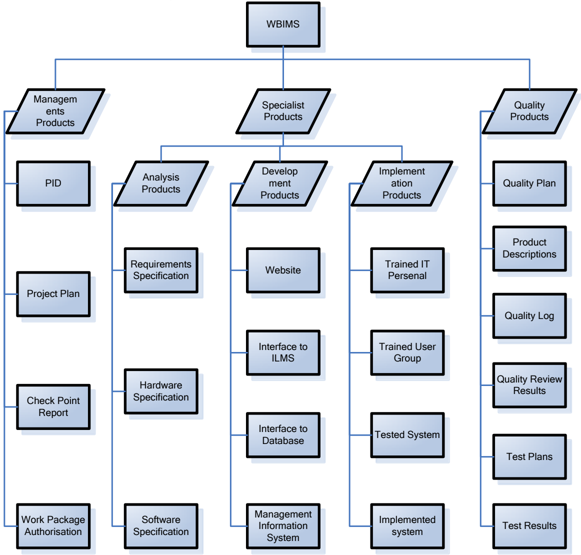


Figure 2. Product Breakdown Structure.

At the bottom level, the individual product is represented by a rectangle shape. A project product is broken down further into one or several activities. The estimate of each activity is derived based on human judgment from the product estimate and the relative complexity of each activity. Again, the total estimated effort for the activities of a product should be equal to the product estimated effort.

Once the PBS is completed, a complete list of the products in that project will be generated. It is time to consider the work of creating a PFD. The principle is that the products in the relation to each other will be looked at and considered how one product is transformed into another. Each product may be consisting of one or more activities. Thus, the activities implied in the delivery of each of the products and those required to create or change the planned products need to be identified to give a fuller picture of the plan’s workload. Figure 3 is an illustration of adding the activities and dependencies based on the PBS of WBIMS.

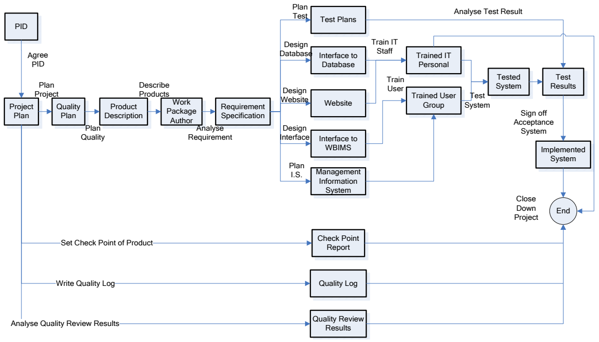


Figure 3. Project Flow Diagram.

Furthermore, the basic configuration of the system requires the entry of top level information about the project such as project category, project timeline, work package description, tolerance level, customer, etc.

C. Product-Based Project Portfolio

A product-based project portfolio (PBPP), as shown in Table I, was proposed in this research to contain more detailed information of each product apart from time, cost, resource and dependencies, such as quality criteria, constrains and activities underneath.

TABLE I. PROJECT PORTFOLIOS

	Product based project portfolio
1	Product name and description
2	Duration of completion
3	Man power
4	Cost including labour & material
5	Dependences & pre-requisites
6	Activities undertaken of each product include details of rework
7	Quality assessment criteria
8	Special technical requirements
9	Constrains & inheritable risks

The PBPP is a top level methodology to use the product-based approach for portfolio collection, project planning and project delivery. It details processes starting from product breakdown until the resource arrangements during the planning stage. The input of PBPP is from information collected from all completed projects and the output is to the new projects. The data repository of PBPP contains both project and simple product data. When a project manager plans a new project, the first step is to break the project into simple products by using PBS and PFD, then PBPP will be looked at and the portfolio of previous projects and simple products can be accessed to see whether those products have been done before. As long as the simple products are found as the same or similar, the benchmarking principle can be employed to choose a suitable product. The activities associated with the chosen product will be regarded as the most suitable practices to deliver the product in the new project plan.

Sometimes project managers need to estimate the time and cost according to their experience when there is no information found from the system. Actual information collected during the project delivery can be stored into the PBPP system again to cross check the accuracy of the previous planning to improve the calculation method for future references.

The PBPP has the obvious advantages to allow maximum information and best practice sharing among projects at the product level. It overcomes the limitation of traditional activity based methods when sharing information at the activity level.

D. Product Portfolio Benchmarking - Automomous Process

After breaking the final product into sub-products, the next step is to benchmark the sub-products to find the best practice among the historic data for project planning. The Quartile approach [14] was applied here to enhance the benchmark process. It shows the spread of the most popular representatives for non-numerical data. This concept refers to the subset of all data values in each of those parts.

TABLE II. QUARTILE LEVELS

User Requirement			
Time (Prior)		**Cost**	
Quartile	**Criteria**	**Quartile**	**Criteria**
Maximum	1	Maximum	**1**
Upper Quartile	**0.75**	Upper Quartile	0.75
Medium	0.5	Medium	0.5
Lower Quartile	0.25	Lower Quartile	0.25
Minimum	0	Minimum	0

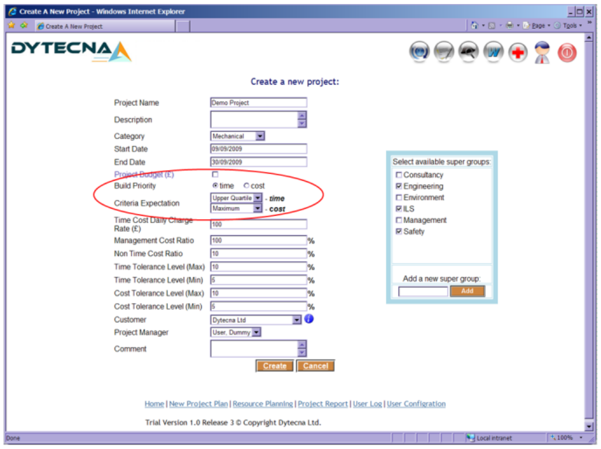
In terms of the user requirements, the WBIMS project delivery time is crucial to the customer, for this reason, the project completion date was fixed; thus, the project time chosen as the higher Build Priority of the project when using iPAS software. In other words, the project completion Time is prior to Cost during the benchmarking process, the Criteria Expectation of project completion Time was set as Upper Quartile and project completion Cost was set as Maximum (see Table П) when configuring the project settings in iPAS (Figure 4), which means all the sub-products in WBIMS project will be benchmarked by following steps:

1) the top 25% products of all products in data repository that have a fairly good completion time will be selected firstly,

2) these selected products will be put into the second round selection to meet the lower prior criteria - the selected

product must have the lowest cost among those completed products,

3) the product that meets both criteria on Time and Cost will be regarded as the best practice, the completion Time and spent Cost as well as all activities that came with this product will be used for generating a new project plan.



Create a new project:

Project Name	Demo Project
Description	
Category	Mechanical
Start Date	09/09/2009
End Date	30/09/2009
Project Budget (£)	□
Build Priority	● time ○ cost
Criteria Expectation	Upper Quartile - *time* Maximum - *cost*
Time Cost Daily Charge Rate (£)	100
Management Cost Ratio	100 %
Non Time Cost Ratio	10 %
Time Tolerance Level (Max)	10 %
Time Tolerance Level (Min)	5 %
Cost Tolerance Level (Max)	10 %
Cost Tolerance Level (Min)	5 %
Customer	Dytecna Ltd
Project Manager	User, Dummy
Comment	

Create Cancel

Select available super groups:
□ Consultancy
☑ Engineering
□ Environment
☑ ILS
□ Management
☑ Safety

Add a new super group: Add

Home | New Project Plan | Resource Planning | Project Report | User Log | User Configration

Trial Version 1.0 Release 3 © Copyright Dytecna Ltd.

Figure 4. Project Flow Diagram.

The formula below for locating the position of the observation at a given percentile, y, with n data points sorted in ascending order is: $L_y = n \cdot \frac{y}{100}$.

Case 1: If L is a whole number, then the value will be found halfway between positions L and L+1.

Case 2: If L is a decimal, round up to the nearest whole number. (for example, L = 1.2 becomes 1).

Through the formula above and Weighted Mean [15] formula, benchmarked values (Time and Cost) of products can be worked out. Together with the portfolio details such as activities, dependencies and constrains obtained from an identified product from data repository, all these information will be used and stored to create a new product portfolio in WBIMS project. As long as all simple products in the WBIMS project are found as the same or similar to the previous completed products, the benchmarking principle can be employed to choose suitable products. The activities associated with the chosen product will be regarded as the most suitable practices to deliver the product in the new project plan. Actual information collected during this project delivery were stored into the portfolio system to cross check the accuracy of the previous planning to improve the calculation method for future references.

E. System Implementation

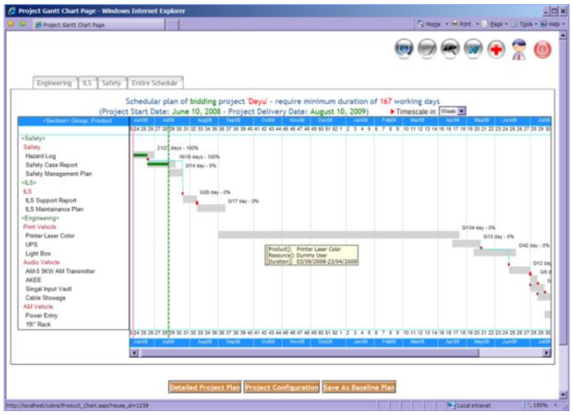
The iPAS system was developed using the latest ASP.NET technology and deployed in Microsoft Windows Server 2008 and MS SQL Server 2008R2 under the Windows environment; it takes advantage of many features of the .NET framework 4.0, such as the SQL data source API, integrated AJAX support, Web Services, and a security model that protects data even in Internet applications.

IV. OVERVIEW OF IPAS FUNCTIONS

As a web-based project management system, iPAS is able to intelligently support project managers in project planning, optimising business performance and project cost. The other main facilities provided by the system are: reverse planning, resource allocation, project monitoring and project reporting. Each of the facilities will be introduced next.

A. Project Planning

iPAS enables project managers to plan a project by following pre-defined products (or work packages). It is also the key step of the product-based planning technique in PRINCE2, which has emerged based upon the idea of considering the products that will result from the project rather than how to execute the work [5]. Apart from creating a project plan without applying historical data, users can create a new project plan benchmarking from previous practices and applying desired criteria. In terms of the chosen category of the new project previously matched historical projects will be listed; the users are able to choose the most desirable project(s) from the list to clone. As long as the products together with their associated activities are selected from the desirable project(s) and submitted for assembling; the portfolios (e.g., product name, activity name, dependencies and feedback) will be copied cross to the new project. The effort (time and cost) of each activity will be calculated based on the customised benchmarking criteria and benchmarking algorithms. As a result, a new project Gantt chart (shown in Figure 5) will be generated according to the time effort.



Schedular plan of bidding project 'Deyu' - require minimum duration of 167 working days
(Project Start Date: June 10, 2008 - Project Delivery Date: August 10, 2009)
Timescale in: Week

<Safety>
Safety
Hazard Log
Safety Case Report
Safety Management Plan
<ILS>
ILS
ILS Support Report
ILS Maintainance Plan
<Engineering>
Print Vehicle
Printer Laser Color
UPS
Light Box
Audio Vehicle
AM-5 5KW AM Transmitter
AKEE
Singal Input Vault
Cable Stowage
AM Vehicle
Power Entry
19\" Rack

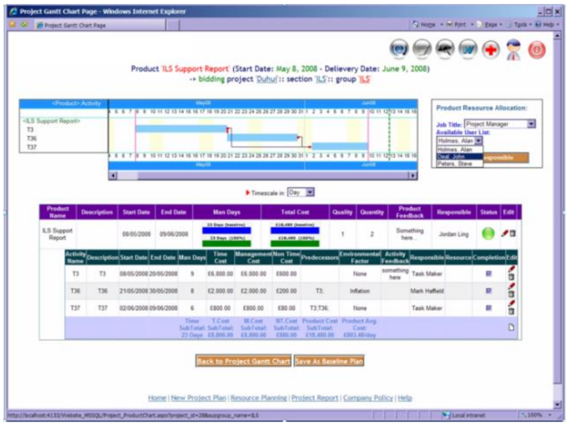
[Product]: Printer Laser Color
[Resource]: Dummy User
[Duration]: 03/09/2008-23/04/2009

Figure 5. A Project Gantt Plan.

Of course, such automatically derived plan allows manual overrides by privileged users for special considerations such as adding new products, removing unnecessary products or editing the statistics of the effort before the project starts. iPAS also enables users to amend the project ending date or start date after the project plan has been generated, the project plan and Gantt chart will automatically adjust to fit the new duration.

B. Human Resource Management and Profiling

The iPAS provides a basic management of staff resource allocation and activity assignment. It has an embedded feature to allow the project manager to authenticate staff's work absences and record the period absent, such as sickness, public holidays and off-site training for all project team members. With the help of this feature, the project manager is able to assign available skilled staff into project products (or work packages) and activities (shown in Figure 6). The data of staff allocation together with other project portfolios stored in the database could be used for generating live project resource allocation reports and other analytical reports.


Project Gantt Chart Page - Windows Internet Explorer
Project Gantt Chart Page
Home
Print
Page
Tools
Help
<Product>-Activity
<ILS Support Report>
T3
T36
T37
May08
Jun08
Timescale in Day
http://localhost:4133/Website_MSSQL/Project_ProductChart.aspx?project_id=28&supgroup_name=ILS
Local intranet
100%

Product 'ILS Support Report' (Start Date: May 8, 2008 - Delievery Date: June 9, 2008)
-» bidding project 'Duhui' :: section 'ILS' :: group 'ILS'

Product Resource Allocation:
Job Title: Project Manager
Available User List: Holmes, Alan
Holmes, Alan
Deal, John
Peters, Steve
Responsible

Product Name	Description	Start Date	End Date	Man Days	Total Cost	Quality	Quantity	Product Feedback	Responsible	Status	Edit
ILS Support Report		08/05/2008	09/06/2008	23 Days (baseline) 23 Days (100%)	£18,400 (baseline) £18,400 (100%)	1	2	Something here...	Jordan Ling	●	

Activity Name	Description	Start Date	End Date	Man Days	Time Cost	Management Cost	Non Time Cost	Predecessors	Environmental Factor	Activity Feedback	Responsible	Resource	Completion	Edit
T3	T3	08/05/2008	20/05/2008	9	£6,000.00	£6,000.00	£600.00		None	something here	Task Maker		☑	
T36	T36	21/05/2008	30/05/2008	8	£2,000.00	£2,000.00	£200.00	T3;	Inflation		Mark Hatfield		☑	
T37	T37	02/06/2008	09/06/2008	6	£800.00	£800.00	£80.00	T3;T36;	None		Task Maker		☑	
				Time SubTotal: 23 Days	T.Cost SubTotal: £8,800.00	M.Cost SubTotal: £8,800.00	NT.Cost SubTotal: £880.00	Product Cost SubTotal: £18,400.00	Product Avg. Cost: £803.48/day					

Back to Project Gantt Chart | Save As Baseline Plan

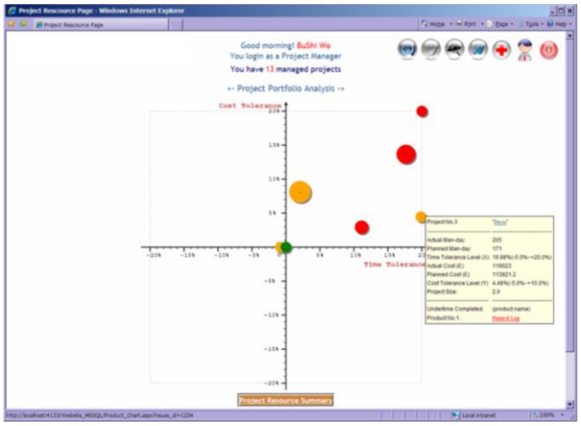
Home | New Project Plan | Resource Planning | Project Report | Company Policy | Help

Figure 6. A Product with Activities.

C. Project Monitoring and Alert Mechanism

iPAS provides a mechanism to automatically monitor and analyse product effort values and work completion status during the project progress according to project baseline. This mechanism depends on the regularly entering the actual effort spent by each person or team assigned to the specific activity as soon as that specific activity is completed. The responsible person is also required to enter real effort to complete a task and to comment on environmental factors affecting the delivery result. When the completion box of an activity is ticked, the activity is considered completed. Since activities are associated to products, actual effort can be summarised at product level and even at project level.

Senior members of the project such as project managers are able to check the progress status of all current running projects immediately through a project tolerance Grid chart (shown in Figure 6). This chart provides a project alerting mechanism. There are two levels of alerting mechanism in iPAS: one is at project level and one is at product level. During the project progress, if the position of a project is inside the tolerance level frame but may be over time, over budget or both; the bubble colour will be shown as amber and means the project is still under control but needs to be carefully monitored. The project manager is expected to analyse the problem or look for extra resources. If the position of a project is outside the tolerance level frame, the bubble colour will be shown as red meaning it is beyond the project tolerance level. This situation requires an exception plan to be launched in accordance with PRINCE2 processes. The project bubble colour will be shown as green if the project is on time and on budget. From this Grid view, user (dependent on privileges) is also able to click through the link of the product and find more details in a product view. For each product, there is also a status traffic light indicator designed for the project manager to understand what is due, what is completed and what is overdue (shown in Figure 7).



Good morning! BuShi Wo
You login as a Project Manager
You have 13 managed projects
<- Project Portfolio Analysis ->

Project No.3	'Deus'
Actual Man-day:	205
Planned Man-day:	171
Time Tolerance Level (X):	19.88%(-5.0%~+20.0%)
Actual Cost (£):	119023
Planned Cost (£):	113921.2
Cost Tolerance Level (Y):	4.48%(-5.0%~+10.0%)
Project Size:	2.0
Undertime Completed:	(product name)
Product No.1:	Hazard Log

Project Resource Summary

Figure 7. Project Tolerance Grid.

D. Project Report

iPAS is able to generate different kinds of reports with charts according to customer requirements. These reports demonstrate project performance, cost analysis, trend analysis, resource allocation and real-time project status, etc. All these reports can be exported into various formats such as PDF, Microsoft Excel and Word.

E. Summary

Getting everyone consistently using the product based planning method and sharing project information across entire project team and organisation is not easy. iPAS has been developed to bridge the gap between PRINCE2 main principles and its application, providing the user with automated planning, monitoring, reports and human resource allocation. iPAS allows configurable access levels based on roles and rights granted that allow users to access the various management levels and features of the solution based on their individual needs. This approach ensures that each user need only see the functionality and information necessary to perform their responsibilities, thereby making the application easier to use for all stakeholders. iPAS also provides a complete project central database, storing all project data in one location for easy access, saving time and resources. It has built in deliverables' reviews and authorisations are granted online for multi-level granularity cooperation, and progress is updated in real time to reduce the need for costly meetings and expensive time wasting. Accessed across network or intranet, all project staff can share real time project information, best practices and learn from previous experiences with projects; all these enable more accurate future estimating and planning.

In addition, iPAS was designed generically, thus it can be widely used for different industry such as manufacture, education, medicine, construction and rail industries, etc.

The report formats can also be customised according to the requirements from specific users.

V. Evaluations

Compared with the existing project management applications on the market, iPAS integrates project planning activities with product based planning and automated effort estimation in light of user's criteria. This is a more sophisticated project plan method, which is designed to efficiently support plan creation and adjustment online based on the practices from historical data. With this method, iPAS offers a better guidance to project managers even program managers, because it can help in shaping the plan and a breakdown of global project effort estimates into product and activity efforts, tracking project progress with alert mechanisms, ensuring that the project will meet its goals in terms of PRINCE2 main principles. iPAS also takes advantage of this fact by gathering statistics, which provides assistance during project management. In general, iPAS is specifically designed for managing projects following a well-defined principle, which is typical in engineering projects (e.g., software, electrical, mechanical and construction). iPAS has been tested and validated mainly by a few case studies in manufacturing industry domain and scientific research domain. Major benefits were observed right from a case by six project managers and four domain experts from Dytecna, where a four years project was set under control and transformed into a success by researching the goals which established at the creation of the product based project plan. The iPAS has also been used in two research projects in UK's National Physical Laboratory (NPL), which followed the PRINCE principle in the organisation. It has successfully assisted the program officer to plan the projects with available resources and monitor the progress from start to end. During case studies, The iPAS was applied to help project managers to share the project knowledge for generating project plan and controlling project progresses via product based benchmarking.

Although it is difficult to quantitatively assess exactly how much time and cost were saved for project planning and management in the case study, users who have used the iPAS system summarise the following major advantages against the traditional project management method:

- It allows the company to continuously improve both bidding, planning and project management as well as reduce risk
- It is a novel approach to store and share information among different projects
- It is an innovative method to integrate PRINCE2 and benchmarking principles
- It reduces project starting up and initiation time, reduces management costs by limiting the number of project meetings conducted
- It wins more work for a customer by providing accurate rather than estimated information on costs and duration at tender stage. Thus, a company has more confidence in the accuracy at ITT (Invitation to Tender) responses, customers have more confidence in bids made and associated cost profiles
- Company is able to justify through life costs and plan resources to serve contracts, thus to improve company success and profitability
- Company has continuous improvement in data accuracy providing early identification of the program that is moving toward an adverse situation
- It's adaptable to any other sectors such as construction, rail industries, health services or government, etc.

TABLE III. Culture Changes before and After Using iPAS

Stages	Pre iPAS	Post iPAS
Bidding	Ad hoc and configuration No historical data, estimation based on expert judgment No follow up, no lessons learned	Historical data are available to improve estimation Resonation and improvements of the process
Planning	Activity based planning Last minute identification of the activities	Product based planning, activities can be referred from best practice
Monitoring	Difficult to follow the evolution of a activity or to assess the quality of the completed work	Easily to monitor the progress of the project by watching the delivery quality of products and practices underneath
Control	Hard to know the failure reasons from project team level and response immediately	Failure point can be easily spot out and then take necessary action quickly

Table III shows the culture changes observed using iPAS software in NPL and Dytecna. Before introducing iPAS tool, the project plans were generated based on experts experience probably, in most cases, the lessons learned from previous projects are easily forgotten, and project activities were planned without a clear idea because there is no precise intention on what is going to be delivered. By recording what was done and how much effort was spent, project team members could now easily monitor and control the project progress, accurately assess what they were doing from an objective perspective, as well as learn the lessons from the past.

VI. Conclusion and Further Work

iPAS is a web-based project planning and management tool adapted to the product based planning techniques of PRINCE2, which can be applied to standardised industries such as Construction, Logistic Support, Electronic and Mechanical Systems Engineering. It guides the project manager by recycling historical data and best practices to estimate project resources and to cascade this into manageable products. iPAS provides focus on project objectives, by structuring a plan based on products, by facilitating accurate monitoring of these products throughout the project. These features assist the project team to maintain programmed activities and to meet contract commitments, whilst reducing the management overhead. It can be deduced from the above sections that iPAS already is capable of providing considerable added value in many areas of project management. However, it has become clear to the development team that there are several ways in which iPAS could be further enhanced.

A possible extension could be to link iPAS database with an organisation's host database. As iPAS collects more and more business practice data from a variety of organisations, there is a need to establish an appropriate knowledge base centre. An external benchmarking comparison service also could be provided in order to coordinate with the unique company database system and bring in external knowledge, which will enable the customer to manage the business more efficiently. The future work can also be focused on enhancing human resource management, enhancing the user interface, perfecting the navigation and strengthening the statistical robustness of the system, etc.

REFERENCES

[1] N. Keith, “Knowledge Transfer Partnerships – Dytecna Grant application and proposal report” pp. 30-40, April 2010.

[2] J. V. Hickey, and C. A. Brosna, “Evaluation of Health Care Quality in Advanced Practice Nursing” 2010, pp. 223-229, Springer Publishing Company, ISBN: 0826107664, 9780826107664

[3] Y. S. Tang, “Product Based Project Portfolio (PBPP) for Best Practice and Information Sharing” Journal of Global Business Development June 2009, 1(1): pp. 51-55.

[4] PRINCE2 “Managing Successful Projects with PRINCE2” (2009 by Great Britain: Office of Office of Government Commerce) - ISBN: 9780113310593

[5] MindManager: MindManager [Online]. Available from: http://www.mindjet.com/mindmanager/platforms/ 2015.01.15

[6] ATTASK [Online]. Available from: http://www.attask.com/ 2015.01.15

[7] ASTA Power Project [Online]. Available from: http://www.astadev.com/products/asta-powerproject/ 2015.01.15

[8] Instant Business Network (IBN): [Online]. Available from: http://www.mediachase.com/ibn/projectmanagement.aspx/ 2015.01.15

[9] V. R. Basili and D. M.Weiss, “A Methodology for Collecting Valid Software Engineering Data”. IEEE Transactions on Software Engineering, 10(6), Nov 1984, pp. 728-738.

[10] C. M. Lott, and H. D. Rombach, “Measurement-based guidance of software projects using explicit project plans”. Information and Software Technology, June/July 1993, pp. 356-357: 407C419,

[11] G. Adams, AW. Koop, and S. Mankotia, “Systems and methods for facilitating and managing business projects” - US Patent 7,694,270, 2010

[12] C. Bentley, “Practical PRINCE2” Third Edition, ISBN 0-11-703544-0, published by TSO (The Stationery Office) UK

[13] K. Harold, “Project Management: A Systems Approach to Planning, Scheduling, and Controlling”, 2003, pp: 323-324, 423-424, 8th Ed., Wiley. ISBN 0-471-22577-0.

[14] S. Goswami, A. Chakrabarti, “Quartile Clustering: A quartile based technique for Generating Meaningful Clusters” ", Journal of Computing, volume 4, issue 2, Feb 2012, pp:50-54, ISSN 2151-9617

[15] D. E. Knuth, “The Art of Computer Programming”, Seminumerical Algorithms, volume 2, chapter 4.2.2, third edition, 1998, pp: 232-233. Addison-Wesley, Boston

[16] Microsoft Project [Online]. Available from: https://products.office.com/en-us/Project/project-top-features 2015.01.15

[17] Gantt charts [Online]. Available from: http://www.gantt.com/ 2015.01.15

Method for Parameter Adjustment for Automated Visual Inspection of Botteled Liquids

Irina Topalova
Faculty of German Engineering Education and Industrial Management
Technical University Sofia
Sofia, Bulgaria
e-mail: itopalova@abv.bg

***Abstract* — A method for parameter adjustment for automated visual control of bottled liquids is presented, aiming at reducing the execution time of bottling when liquid colors are similar and not easily visible. Edge profile detection is applied to find the transition points where a line fitting algorithm connects them in a line. The obtained short execution time and very good accuracy enable inspection of bottles in a moving condition. The proposed algorithm is tested with blurred images of beer and mineral water bottles, according to real production conditions. The represented method could be applied in any related cases in which the liquid level is not easily visible and the execution time is a crucial component.**

Keywords - visual inspection; image processing; line fitting; execution time.

I. INTRODUCTION

There are many automated systems for automated visual inspection of the liquid level in the bottling industry [1][2][3][4][5]. The specifics of this production demand liquid level control in the moving condition of the conveyer belt. In the case of sparkle liquids and beer bottled production, one problem is fixing the fill level of foamed surfaces in moving condition, all within a short execution time, even when the bottle and liquid colors are very similar. Thus, the optimization of the decisive algorithm parameters in terms of quick-operation and high accuracy is imperative. In addition, the problem has to be solved with a simple and inexpensive technology equipment, aiming at reducing the production costs.

In this research, a method is proposed for liquid level inspection in moving condition, when the bottle and liquid colors are very similar and the transition between them is not easily visible. Edge profile detection is applied to find the transition points where a line fitting algorithm connects them in a line. The algorithm's parameters are adjusted to minimize execution time, based on the analysis of the influence of the significant image parameters over the execution time. The benefits of the obtained short execution time and very good accuracy enable inspection of bottles in moving condition, thus, eliminating the need of additional technological appliances applying a single smart camera.

The experiments are implemented using a Smart Camera NI 1742. Triggered infrared lighting is used to eliminate the variations in environmental lighting. To simulate the blur noise added to the images because of the conveyer belt movement, the calculated blur for typical conveyer belt velocities in number of pixels is added to each image. Further tests with cameras having different image resolution, by different light intensities, are foreseen.

Section I-A describes the state-of-the-art. In Section II, the overall proposed method for liquid level detection is defined. Section III-A describes the image parameters that influence the execution time/accuracy. Line fitting algorithm is represented in Section III-B. In Section IV, the developed algorithm for parameter adjustment is explained, after the analysis of the execution time. The experiments and the obtained results for images, resembling the moving conveyer belt conditions for many examples are represented and discussed in Section V.

A. State-of-the-Art

Machine vision is implemented nowadays in the modern automated production systems for real-time control of different product parameters [1][3][6]. In automated bottle filling production, most of the checks are concentrated on the presence/absence, position or quality of different bottle parts, such as cap, label or defects. Liquid fill level control is relatively rarely accomplished, especially when the bottle and liquid colors are very similar. For example, the *Q Check* verification system [4] inspects flat and sipper caps on beverage bottles for cap presence and height, dust cap presence and fill level. Because the check is in moving condition, the liquid surface is often falsly recognized and the bottle is incorrectly automatically rejected. *Mettler Toledo system* [5] demonstrates a Full Bottle Inspection system (FBI) with simple part setup, very intuitive train tools with training in less than one minute, rejection off the line of all defective bottles. It checks the liquid level in movement, but without discussion and recommendations about achieved accuracy when fixing the fill level of foamed surfaces. *DATALOGIC* [6] is a system using a complex multiple cameras/mirrors structure for cap and label detection and defective rejection. It represents no fill level control, thereby the rest of the bottle components are checked with fixed parameter values with no discussion about the execution time. In some systems [2], this control is completed in a stop conveyor belt condition, adding to the production line a technological appliance. In this case, the technological cycle time increases together with raising the cost of the system.

In conclusion, we did not succeed to find in the existing similar systems any analysis of the influence of the significant image parameters over the execution time for foamed liquid level determination, while maintaining accuracy. As result of the implemented time analysis, a method for parameter adjustment is proposed. The proposed method is verified with many blurred images, aiming to simulate the real production conditions. The experiments are implemented using only simple machine vision components with no need of adding to the production line some special technological appliances. The main advantage of the presented approach is the non-intuitive, but based on the image parameter analysis method for training the vision system for fast real-time execution. It could be applied in any related case where the liquid level is not easily visible and the execution time is a crucial component.

II. METHOD FOR LIQUID LEVEL LINE DETECTION

The proposed method is based on analyzing the edge strength profiles along different parallel, preliminary fixed lines in a search direction. The algorithm finds the first pixel along each edge strength profile having more than a minimum chosen difference between the intensity values of the edge and the surrounding pixels. The method of least squares is used to determine the best fit line to the data set, formed by detected pixels along all lines. The influence of four parameters – *edge strength, kernel size, projection width* and *interline gap* over the execution time and level line accuracy are analyzed and a method for their adjustment is proposed. The method and its algorithm are tested on 30 samples of brown bottles of beer and 30 samples of white bottles of mineral water. Infrared triggered lighting is used for image acquisition of moving bottles.

III. DEFINITION OF PARAMETERS AND LINE FITTING ALGORITHM

Four parameters – *edge strength, kernel size, projection width* and *interline gap* have the strongest influence over the execution time and accuracy when determining the liquid level line. They are used to detect the liquid level edge points.

A. Definition of Used Parameters

Edge strength – This is the edge contrast. It determines the variation in the grayscale values between the background and the edge. Figure 1 shows the Grayscale profile in a search direction. The edge strength can vary for the changes in lighting conditions. That is reason to use infrared triggered lighting on the acquisition moment to eliminate these changes. The edge length characterizes the slope of the edge. Edges with gradual transitions between the background and the edge have a longer edge length.

Kernel size - A kernel is usually a 3x3, 5x5, 7x7, etc. structure that represents a pixel and its relationship to the pixel neighbors [7]. The chosen size of the kernel should be based on the expected sharpness, or slope of the searched edge.

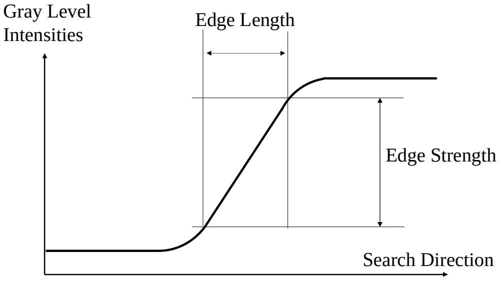


Figure 1. Greyscale profile

Projection width – Determines the amount of pixels perpendicular to the search direction [7], that are averaged at each pixel along the search line to calculate the edge profile strength. The projection width has to be increased when the image is noisy or blured because of the movements of the aquisiting object.

Interline gap – Defines the distance between two neighboring search lines in pixels.

B. Line Fitting

The algorithm finds the first pixel along each edge strength profile having less than a minimum chosen difference/threshold between the intensity values of the edge and the surrounding pixels. All such pixels are considered to be Liquid Level Pixels (LLP) or border pixels. The method of least squares is used to determine the best fit line to the LLPs data set, formed by the detected pixels along all lines. When n LLPs with coordinates [x_i, y_i] are found, the approximating straight line will have the equation

$$Y = f(x) = \alpha_0 + \alpha_1 X . \quad (1)$$

Then, α_0 and α_1 values are searched, so that a minimum Mean Square Distance (MSD), according to Figure 2, will be obtained.

$$MSD = \min_{\alpha_0,\alpha_1} \sum_{i=1}^{n} d_i^2 \quad (2)$$

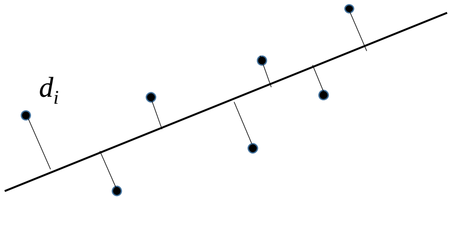


Figure 2. Line fitting

Foreseeing the expressions $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$; $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and

ISBN: 978-1-61208-405-3

$$\sum_{i=1}^{n}\left(x_i-\overline{x}\right)^2=\left(\sum_{i=1}^{n}x_i^2\right)-n.\overline{x}^2=\left(\sum_{i=1}^{n}x_i^2\right)-\frac{1}{n}\left(\sum_{i=1}^{n}x_i\right)^2 \tag{3}$$

the coefficients α_1 and α_0 get the values of

$$\alpha_1=\frac{\left(\sum_{i=1}^{n}x_iy_i\right)-n.\overline{x}.\overline{y}}{\left(\sum_{i=1}^{n}x_i^2\right)-n.\left(\overline{x}^2\right)} \tag{4}$$

$$\alpha_0=\overline{y}-\alpha_1.\overline{x} \tag{5}$$

The line with the best quality is the line that shows the lowest MSD [8]. The quality of the line is further improved by successively removing the furthest pixels from the current line until a preliminary minimum score is obtained.

The result of the line fitting algorithm is a line that is fit to the best set of the LLPs after ignoring the outlying pixels.

IV. Parameter Adjustment

The inspection of the liquid level in motion condition sets requirements of short execution times. The phase of detection of the liquid level, together with the phase of image acquisition, are the most time consuming steps in the algorithm.

A. *Execution Time Analysis*

As the parameter values are decisive for accuracy of liquid line determination, it is important to analyze the influence of the four above mentioned parameters over the execution time. On the base of analyses, optimum proportion parameter values and execution time have to be found. The two graphics in Figures 3 and 4 show that the execution time needed for liquid level detection, including edge detection and line fitting, increases linear with increasing the kernel size and the projection width values. However, the increase in the projection width essentially influences the execution time.

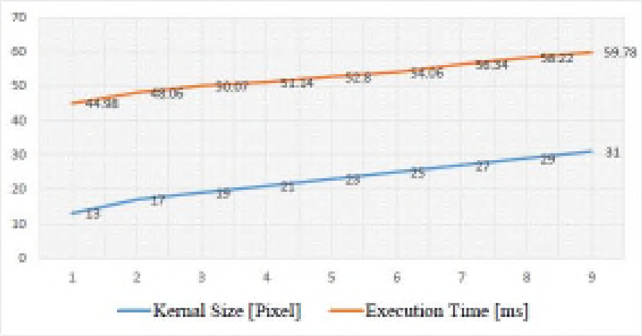


Figure 3. Influence of the Kernel size variations over the Execution time

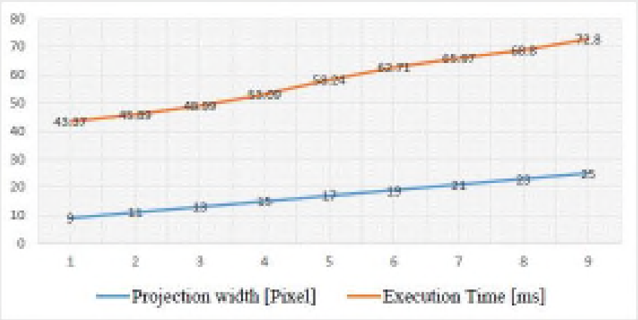


Figure 4. Influence of the Projection width variations over the Execution time

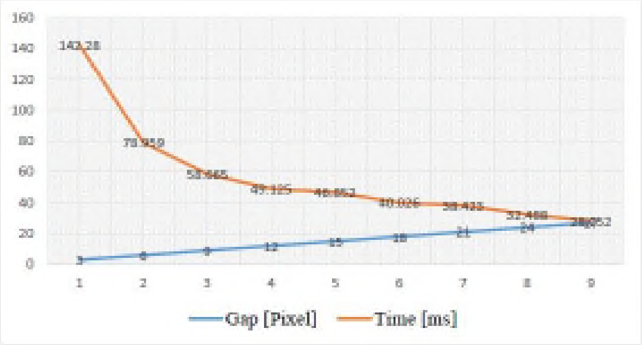


Figure 5. Influence of the Interline gap variations over the Execution time

Figure 5 shows that the increase in the interline gap reduces more rapidly the execution time.

B. *Proposed Method for Parameter Adjustment*

Taking into account that the bottles are inspected in moving condition, obviously some froth is generated, especially in the case of beer production. That means that the intensity along the edge line changes gradually and finding the liquid level edge points requires an increase in the kernel size. Also, it is well known [8] that if the image is noisy, an increase in the projection width is necessary. So, considering these circumstances and aiming at high accuracy, it is reasonable to begin searching the edge profiles with high values of kernel size and projection width and low interline gap value. To reduce the execution time, the following parameter adjustment method is proposed:

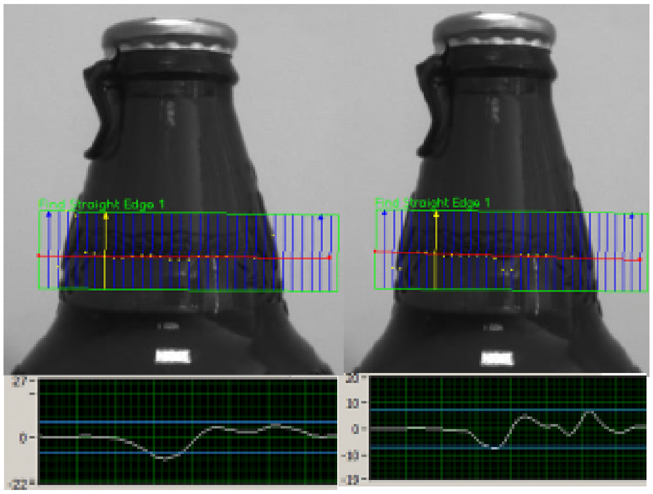
1. Choose high values of kernel size and projection width, choose low values of interline gap to obtain right edge points and right line fitting.
   Straight edge minimum threshold is chosen based on empirical approach.
2. Reduce the kernel size till line fitting is still correct.
3. Reduce the projection width till line fitting is still correct. Stop reducing when line fitting errors appear.
4. Increase interline gap till line fitting is still straight.
5. If no straight edges are found, reduce the straight edge minimum threshold until the step finds a straight edge again.

V. EXPERIMENTS AND RESULTS

The experiments are implemented using a Smart Camera NI 1742 with triggered infrared lighting and software Vision Builder AI 2011. To simulate the blur noise [9] added to the images because of the conveyer belt moving, the calculated blur in number of pixels is added to each image. For a typical conveyer belt velocity of 25m/min ≈ 417 mm/sec and image resolution of 300 dpi ≈ 118,11 dp(cm) ≈ 11,81 dp(mm), the calculated conveyer belt velocity measured in Pixel per second is V_p = 417x11,8 = 4920 Pix/sec. Then, the resulting Motion Blur = V_p * Exposure time = 4920 x 1/125 ≈ 40 Pix Motion Blur. In our case, a short value of Exposure time = 1/500 is chosen which corresponds to 9 Pix Motion Blur. This value is added to the test images to simulate the motion of the bottles [10][11]. Figures 6, 7 and 8 represent the subsequent steps for parameter adjustment and show the change in Edge Strength Profile moving through the steps of the proposed algorithm for parameter adjustment. Finally, the obtained parameter values with line fitting still correct for all of the 60 exemplars are found as: edge strenght 5, kernel size 5, projection width 5 and interline gap 21 pixels. Execution time for 60 exemplars is 60.424 msec.

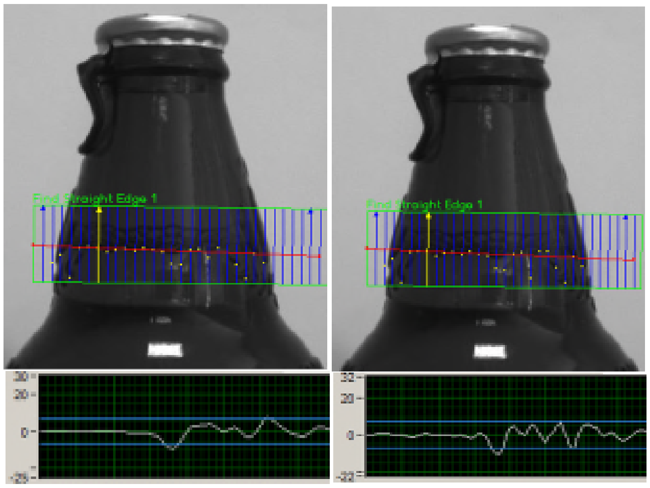
Figure 8 shows the final line fitting with a distinct Edge Strength Profile and an appropriate Minimum Edge Strength/threshold (MES). Figure 10 represents the finally obtained line fitting for some of the tested bottled mineral water samples. Table I. represents the execution time and accuracy for 20, 40 and 60 bottles when moving through the steps of the proposed algorithm.

The accuracy is calculated as [(number of all exemplars - number of exemplars with bad line fitting)/ number of all exemplars].100 [%]. Figure 9 shows the influence of the parameter value reduction over the execution time and over the accuracy according to the data in Table I. The first rising line in the graphic represents cases 1, 2, 3, the second rising line represents cases 4, 5, 6, etc. It is visible that the reduction of parameter *projection width* influences stronger the reduction of execution time (rising lines 3,4, cases 7 to 12) then reduction of kernel size (cases 4, 5, 6). The strongest is the influence of inceasing the interline gap (cases 16, 17, 18).



(a) (b)

Figure 6. Edge Strength Profile for search line 8: (a) edge strenght 7, kernel size 23, projection width 23 and interline gap 9 pixels; (b) edge strenght 7, kernel size 9, projection width 23 and interline gap 9 pixels



(a) (b)

Figure 7. Edge Strength Profile for search line 8: (a) edge strenght 7, kernel size 9, projection width 9 and interline gap 9 pixels; (b) edge strenght 7, kernel size 5, projection width 5 and interline gap 9 pixels

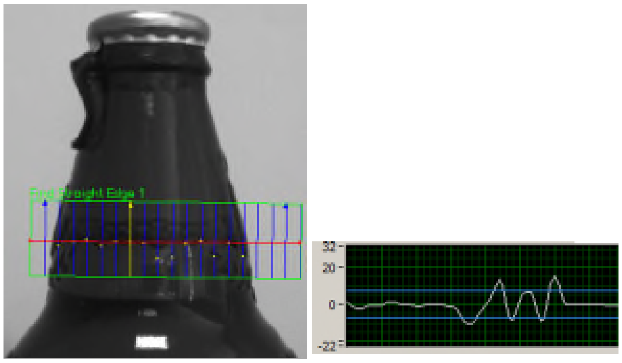


Figure 8. Edge Strength Profile for search line 8: edge strenght 5, kernel size 5, projection width 5 and interline gap 21 pixel

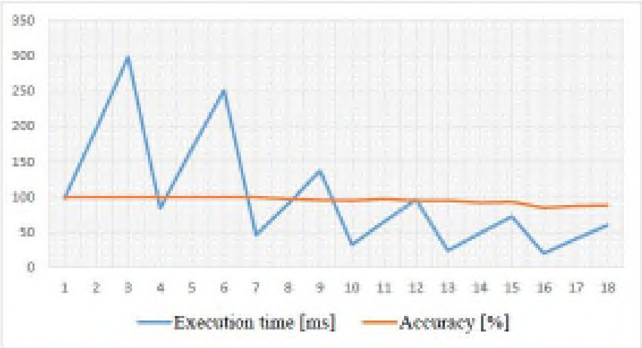


Figure 9. Execution time and accuracy for 20, 40 and 60 tested exemplars according to the cases 1 to 18 in Table I.

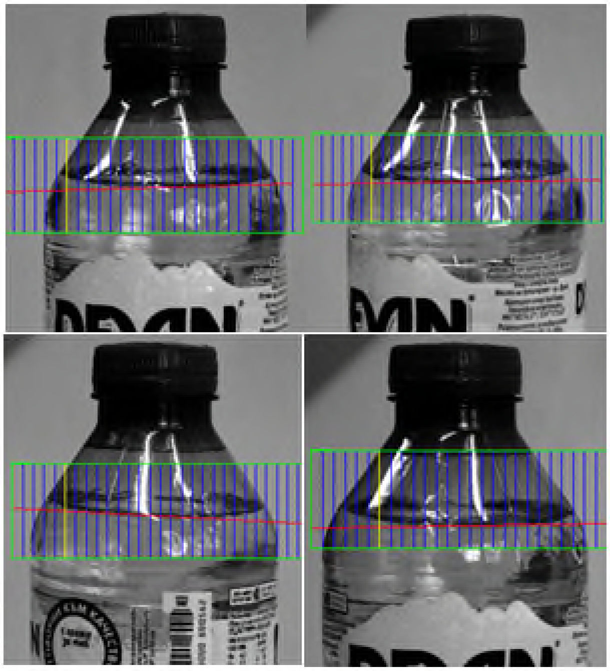


Figure 10. Line fitting for bottled mineral water - edge strenght 32, kernel size 9, projection width 9 and interline gap 77 pixels

TABLE I. EXECUTION TIME AND ACCURACY

Line fitting parameters	*Case*	*Test Samples*	*Time[ms]*	*Accuracy[%]*

MES=7; Gap = 9
Kernel Size = 23;
Projection Width = 23
	1	20 bottles	97.642	100.00

	2	40 bottles	196.462	100.00
	3	60 bottles	297.709	100.00

MES=7; Gap = 9
Kernel Size = 9;
Projection Width = 23
	4	20 bottles	84.176	100.00

	5	40 bottles	168.692	100.00
	6	60 bottles	250.847	100.00

MES=7; Gap = 9
Kernel Size = 9;
Projection Width = 9
	7	20 bottles	46.216	100.00

	8	40 bottles	90.418	97.50
	9	60 bottles	137.102	96.00

MES=7; Gap = 9
Kernel Size = 5;
Projection Width = 5
	10	20 bottles	32.941	95.00

	11	40 bottles	64.988	97.50
	12	60 bottles	96.494	95.00

MES=7; Gap = 15
Kernel Size = 5;
Projection Width = 5
	13	20 bottles	24.434	95.00

	14	40 bottles	48.811	92.50
	15	60 bottles	72.461	93.30

MES=5; Gap = 21
Kernel Size = 5;
Projection Width = 5
	16	20 bottles	20.431	85.00

	17	40 bottles	41.047	87.50
	18	60 bottles	60.424	88.30

The optimum execution time reduction is obtained and the parameter value adjustment stops when accuracy falls between 88.5% and 88.3%, because further parameter adjustments will reduce the obtained accuracy.

VI. CONCLUSION

The obtained results show that the proposed method for liquid level inspection with parameter adjustment is suitable even when the bottle and liquid colors are very similar and the transition between them is not properly visible. It was tested with blurred images to simulate the conveyer belt movement in real production. The experiments are implemented using only simple machine vision components with no need of adding to the production line some special technological appliances. The main advantage of the represented approach is non-intuitive, but based on the image parameter analysis method for training the vision system for fast real-time execution. The represented method could be applied in any related task where the execution time is a crucial component. In order to generalize this method, further tests with cameras having different image resolution, by different light intensities, are possible. Although having in mind that the most up-to-date automated visual systems use triggered infrared lighting, we expect these variations will not impact significantly the proposed methodology.

REFERENCES

[1] http://www.fdbusiness.com/2013/11/inspection-system-improves-productivity-in-beer-industry/, retrieved: April, 2015.

[2] https://www.youtube.com/watch?v=_7pkjlEmzqI, Visual system developed by National Instruments, retrieved: April, 2015.

[3] http://globalinspectiontek.com/_3ce8bb4d.html, retrieved: April, 2015.

[4] https://www.youtube.com/watch?v=ZG9GIQgmisY, retrieved: May, 2015.

[5] https://www.youtube.com/watch?v=BPyrfivjbSE, Metter Toledo System, retrieved: May, 2015.

[6] https://www.youtube.com/watch?v=ffkdAPKeuIU, DATALOGIC Visual System, retrieved: May, 2015.

[7] W. K. Pratt, “Introduction to Digital Image Processing”, CRC Press, ISBN 9781482216691, September 13, 2013.

[8] S. J. Ahn, “Geometric Fitting of Parametric Curves and Surfaces”, Journal of Information Processing Systems 4, December, 2008, pp. 153-158.

[9] B. G. Batchelor, Machine Vision Handbook, Springer, ISBN 9781849961684, 2012.

[10] National Instruments, NI Vision Builder for Automated Inspection User Manual, 373379C-0. National Instruments, 2006.

[11] National Instruments, NI Vision Builder Assistant, 2011.

Estimation of Time to Contact from Blurred Images

Yukitada Takanashi, Kazuyuki Ito
Dept. of Electrical and Electronics Engineering
HOSEI University
Tokyo, Japan
e-mail:10x2061@stu.hosei.ac.jp, ito@hosei.ac.jp

***Abstract*— Recently, intelligent safety systems, such as autonomous collision avoidance for automobiles have attracted considerable attention. In this paper, we propose an algorithm that can estimate time to contact by using blurred images that are captured by a monocular camera rather than distance information. We conducted experiments in order to confirm the validity of the algorithm.**

Keywords-ecological psychology; τ-margin; monocular camera; crush avoidance; blurred image; time to contact

I. INTRODUCTION

Recently, intelligent safety systems, such as autonomous collision avoidance for automobiles have attracted considerable attention. Automobiles are typically equipped with distance sensors or stereo cameras to detect obstacles in their path [1][2].

In conventional works, there are three major methods for measuring distance between the automobile and the obstacle [3]. Table I shows the features of the three major methods.

TABLE I. FEATURES OF THE MAIN METHODS TO MEASURE AUTOMOBILE-OBSTACLE DISTANCE

	Bad weather environment	Dark conditions	Cost
Stereo camera	Not-applicable	Not-applicable	Middle
Laser radar	Not-applicable	Applicable	Low
Millimeter-wave radar	Applicable	Applicable	High

In general, in order to measure distances in dark conditions, the cost to realize the system becomes high because a combination of the multiple methods is required in this case.

On the other hand, in the context of ecological psychology [4][5], it has been demonstrated that time to contact can also be estimated by simply using monocular visual information rather than distance information. In ecological psychology, time to contact is called tau-margin, and it can be calculated based on the apparent size of an approaching object and its temporal change [6].

In our previous studies, we proposed methods to estimate tau-margin using the images of a monocular camera [7]. However, in dark conditions, such as those at night, it was very difficult to estimate tau-margin because of blurred images.

To address this issue, in this paper, we propose an algorithm that can estimate the tau-margin at night despite blurred images acquired from monocular camera.

We conducted experiments in order to confirm the validity of the algorithm.

The rest of the paper is organized as follows. Section II introduces the tau-margin. Section III describes our proposed algorithm tau-margin using blurred images. Section IV verifies the proposed algorithm. Section V concludes this paper.

II. TAU-MARGIN

Figure 1 shows an object approaching a camera.

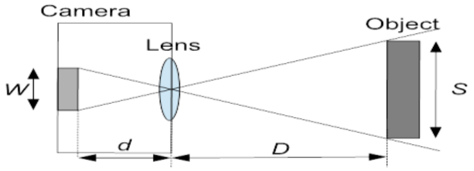


Figure 1. Appearance of the object.

The apparent size W can be expressed by (1). The temporal change $\dot{W}$ is given by (2), where $\dot{D}$ is the approaching speed.

Equation (3) is obtained from (1) and (2). Equation (3) implies that the time to contact $-D/\dot{D}$ is obtained from $W/\dot{W}$. In ecological psychology, $W/\dot{W}$ is called tau-margin (τ).

$$W = \frac{d}{D}S \tag{1}$$

$$\dot{W} = -\frac{dS}{D^2}\dot{D} \tag{2}$$

$$-\frac{D}{\dot{D}} = \frac{W}{\dot{W}}(= \tau) \tag{3}$$

In our previous study [7], we estimated tau-margin based on the movement of each pixel. Figure 2 shows the movement of pixels and Figure 3 shows the coordinate system.

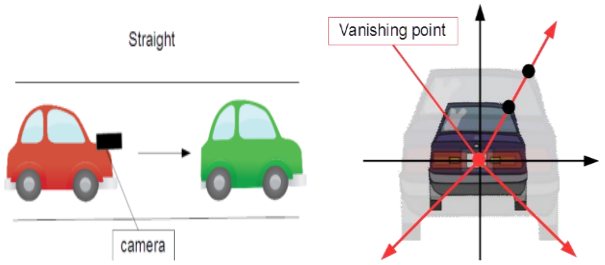


Figure 2. Movement of pixels.

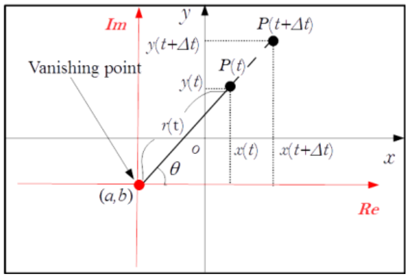


Figure 3. Coordinate system.

The center of the expanding image is called the vanishing point. In Figure 3, the origin of the polar coordinate system is the vanishing point. In the polar coordinate system, the expansion of an image is expressed by (4) and (7). The position of the vanishing point moves when the car turns. The movement of the vanishing point is expressed in the X-Y coordinate system in Figure 3. Thus, the movement of each pixel is given by (5) and (6), and (8) and (9), where Δt is the time interval.

$$P(t+\Delta t) = P(t)\left\{1+\frac{\Delta t}{\tau(t)}\right\} \tag{4}$$

$$x(t+\Delta t) = \{x(t)-a\}\left\{1+\frac{\Delta t}{\tau(t)}\right\}+a \tag{5}$$

$$y(t+\Delta t) = \{y(t)-b\}\left\{1+\frac{\Delta t}{\tau(t)}\right\}+b \tag{6}$$

$$P(t-\Delta t) = P(t)\left\{1+\frac{\Delta t}{\tau(t-\Delta t)}\right\}^{-1} \tag{7}$$

$$x(t-\Delta t) = \{x(t)-a\}\left\{1+\frac{\Delta t}{\tau(t-\Delta t)}\right\}^{-1}+a \tag{8}$$

$$y(t-\Delta t) = \{y(t)-b\}\left\{1+\frac{\Delta t}{\tau(t-\Delta t)}\right\}^{-1}+b \tag{9}$$

III. Proposed Algorithm

Figure 4 shows the setting of camera, and Figures 5-7 show the algorithm.

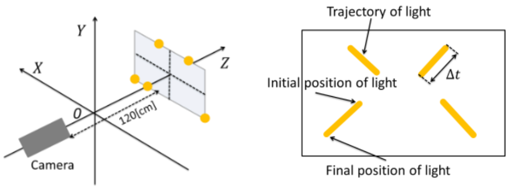


Figure 4. Setting of camera and its blurred image.

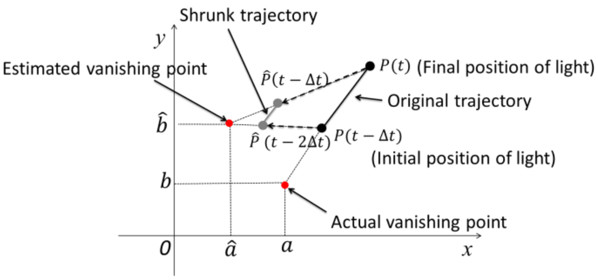


Figure 5. Estimated vanishing point.

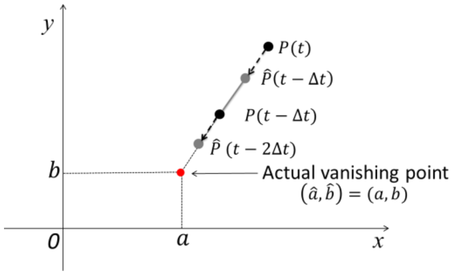


Figure 6. Reduce a locus of light.

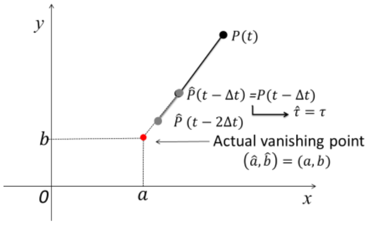


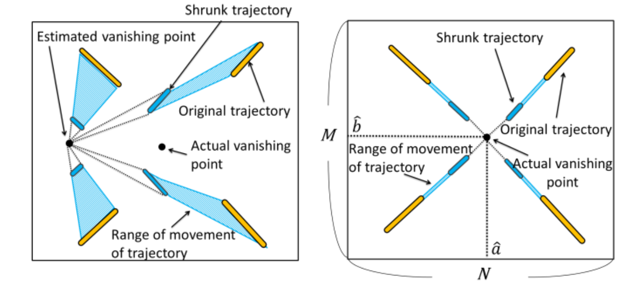
Figure 7. Acquisition of tau-margin.

In this study, we propose an algorithm that can estimate tau-margin using blurred images. Figure 4 shows an example of a blurred image captured in dark conditions. We assume that static point light sources are on the same plane perpendicular to the direction of camera's movement, and the trajectory of the point light source on the captured image is a straight line, as shown in Figure 4. We process the entire image without having to distinguish a point light sources. These trajectories include information on the movement of the moving camera. The inside edge of the trajectory is the initial position of the light and the other side is its final position.

We estimate the vanishing point (a, b) using (8) and (9). First, we shrink the trajectory by substituting $\hat{a}$, $\hat{b}$, and $\hat{\tau}$ in (8) and (9), where $\hat{a}$, $\hat{b}$, and $\hat{\tau}$ are estimated values. Through trajectory shrinking, the trajectory moves to the estimated vanishing point $(\hat{a}, \hat{b})$, as shown in Figures 5-7. As shown in Figure 5, when there is an erroneous position between the actual vanishing point and the estimated vanishing point, the shrunk trajectory does not lie on the original trajectory.

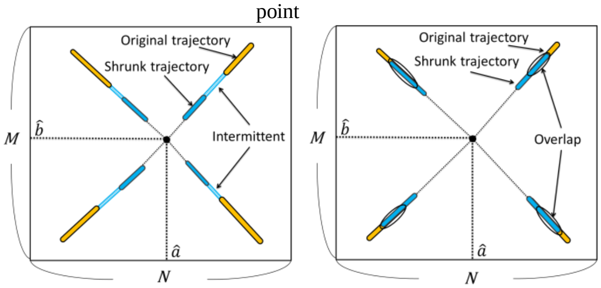
On the other hand, as shown in Figure 6, the estimated vanishing point and the actual vanishing point are the same. The shrunk trajectory moves to the original trajectory towards the actual vanishing point. By conforming the shrunk trajectory to the original trajectory, we can obtain the estimated values of $\hat{a}$, $\hat{b}$, and $\hat{\tau}$.

To estimate $\hat{a}$, $\hat{b}$, and $\hat{\tau}$, we employ the method of least squares. Figure 8 shows changes in the trajectory due to the position of the vanishing point.



(a) Vanishing point has error (b) Vanishing point has no error

Figure 8. Changes in the trajectory due to the position of the vanishing point



(a) Tau is too small (b) Tau is too large

Figure 9. Changes in the trajectory due to the value of tau

When there is an erroneous position between the actual vanishing point and the estimated vanishing point, the trajectory shrinks, as shown in Figure 8 (a). On the other hand, when there is no error, the trajectory shrinks, as shown in Figure 8 (b).

By minimizing the area of the rectangle composed of the original trajectory and the shrunk trajectory, we obtain estimated position of the vanishing point ($\hat{a}$, $\hat{b}$).

In the same way, as shown Figure 9, by minimizing the area of the overlaps and the intermittent between the original trajectory and the shrunk trajectory, we obtain the estimated time to contact $\hat{\tau}$. Figure 10 the flowchart of the theory and Table II defines the parameters used in the flowchart. In Figure 10, we employ the coordinate system in Figure 11.

TABLE II. PARAMETERS

$B(i,j)$	Binary image
$S(i,j)$	Shrink image
$S_a(i,j)$	Accumulation of shrink image
(i_b, j_a)	Position of vanishing point
M	Height of image
N	Width of image
R	Shrink rate
R_{max}	Upper limit of shrink rate
ΔR	Step size of R_{max}
Δt	Shutter speed
est_R	estimate value of R
est_i_b	estimate value of i_b
est_j_a	estimate value of j_a
est_τ	estimate value of time to contact τ
min_S_a	minimum value of sum of pixels of $S_a(i,j)$
min_dis	minimum value of sum of pixels of the overlaps and the intermittent between $B(i,j)$ and $S(i,j)$

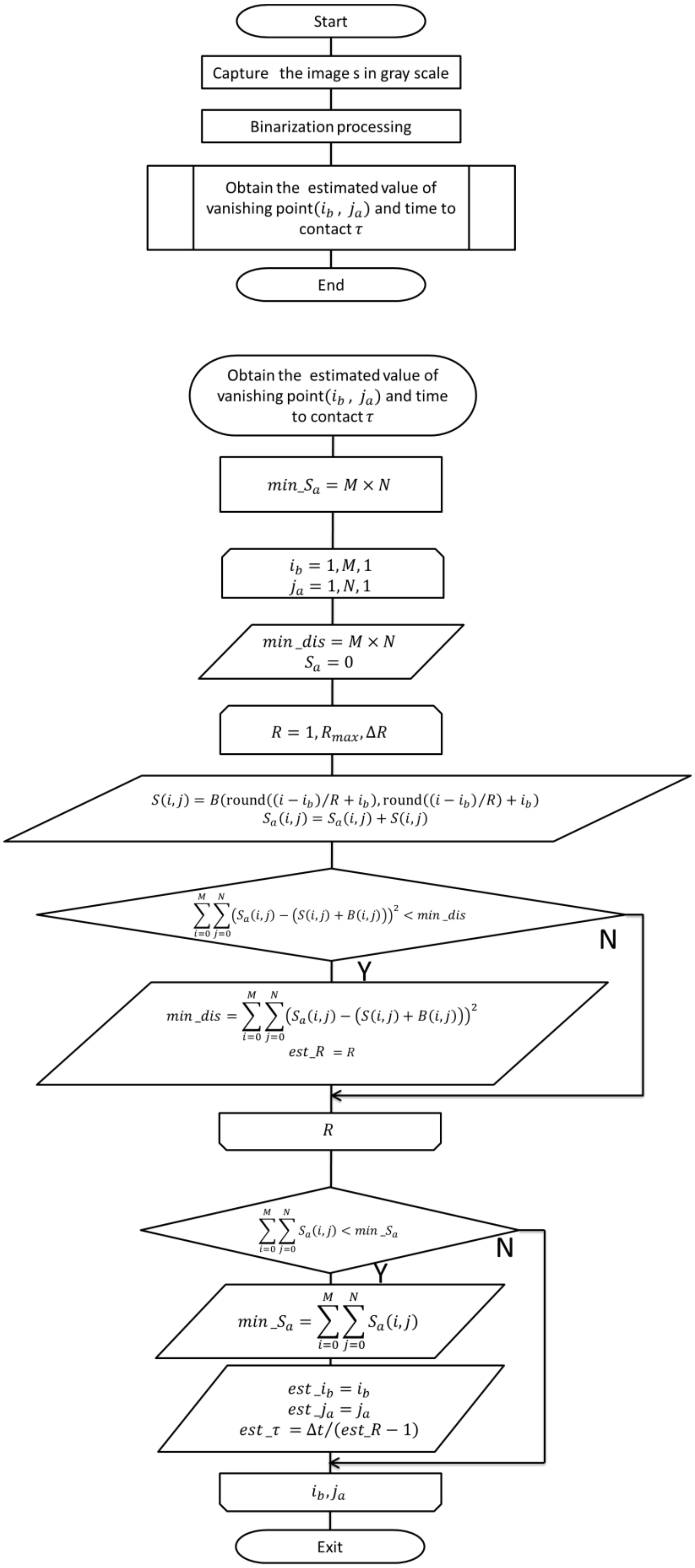

Start
Capture the image s in gray scale
Binarization processing
Obtain the estimated value of vanishing point(i_b , j_a) and time to contact τ
End
Obtain the estimated value of vanishing point(i_b , j_a) and time to contact τ
$min\_S_a = M \times N$
$i_b = 1, M, 1$
$j_a = 1, N, 1$
$min\_dis = M \times N$
$S_a = 0$
$R = 1, R_{max}, \Delta R$
$S(i,j) = B(\text{round}((i - i_b)/R + i_b), \text{round}((i - i_b)/R) + i_b)$
$S_a(i,j) = S_a(i,j) + S(i,j)$
$\sum_{i=0}^{M}\sum_{j=0}^{N}(S_a(i,j) - (S(i,j) + B(i,j)))^2 < min\_dis$
N
Y
$min\_dis = \sum_{i=0}^{M}\sum_{j=0}^{N}(S_a(i,j) - (S(i,j) + B(i,j)))^2$
$est\_R = R$
R
$\sum_{i=0}^{M}\sum_{j=0}^{N} S_a(i,j) < min\_S_a$
N
Y
$min\_S_a = \sum_{i=0}^{M}\sum_{j=0}^{N} S_a(i,j)$
$est\_i_b = i_b$
$est\_j_a = j_a$
$est\_\tau = \Delta t/(est\_R - 1)$
i_b, j_a
Exit

Figure 10. Flowchart of the theory

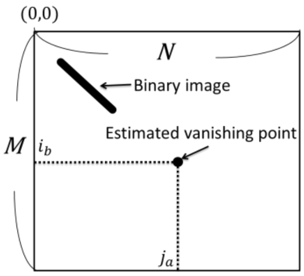


Figure 11. Coordinate system for captured image

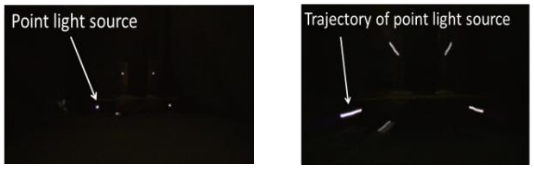
IV. EXPERIMENT

We conducted an experiment to verify the basic capability of the proposed method. Table III shows the setting of the experiment. Processing was conducted offline. Processing time per image was approximately 90 seconds.

TABLE III. SPECS OF THE PC AND EXPERIMENT SETTING

OS	Windows 7 Enterprise
CPU	Intel(R) Core(TM) i3 1.33GHz
Memory	4GB
Application for calculation	MATLAB R2013a
Image size	150×300 [pixel]
Shutter speed of the camera	0.5 [sec]

The camera moved to the point light source by a constant speed, as shown in Figure 4. Figure 12 depicts the captured images. Figure 13 shows the estimated time to contact ($\hat{\tau}$). From Figure 13, we can confirm that the time to contact is successfully estimated.



(a) Still image

(b) Blurred image

Figure 12. Five point light sources.

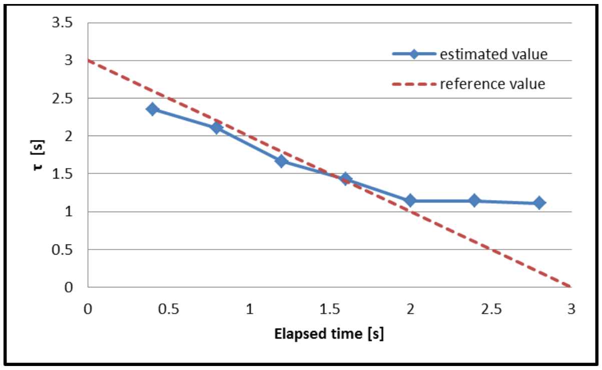


Figure 13. Experiment result.

V. CONCLUSION AND FUTURE WORK

In this paper, we focused on the framework of ecological psychology and we proposed a simple algorithm to estimate the time to contact using blurred images. In this algorithm, expansion of obstacles on captured images is estimated from the trajectories of point light sources on the blurred images, and the time to contact to the obstacles is obtained. Thus, the proposed algorithm is applicable to dark conditions.

To demonstrate the effectiveness of the proposed algorithm, an experiment in a simple dark condition was conducted and we confirmed that time to contact could be estimated.

In the future, we plan to apply the proposed approach to various types of real environment and verify its usability in that environment.

ACKNOWLEDGMENT

This research was partially supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C), No. 24500181.

REFERENCES

[1] I. Joung and I. Ahn, “Two-dimensional depth data measurement using an active omni-directional range sensor,” IEICE Trans. Fund. Electron., Commun. Comp. Sci., vol. E84-A, no. 5, 2001, pp. 1288–1292.

[2] B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision”, Proc 7th Intl Joint Conf on Artificial Intelligence, 1981, pp. 674-679.

[3] Nikkei Electronics:NE Handbook series Sensor Networks: Nikkei Business Publications, Inc., June, 2014, pp. 16-19.

[4] J. J. Gibson, “Reasons for Realism: Essays in Feminist Theory,” Lawrence Erlbaum Associates, 1982.

[5] J. J. Gibson, “The Ecological Approach to Visual Perception,”Lawrence Erlbaum Associates, 1986.

[6] D. N. Lee, “The optic flow field: The foundation of vision,” Phil. Trans. Royal Soc. London B, vol. 290, no. 1038, 1980, pp. 169–179.

[7] Y. Kawai and K. Ito, “Estimation Method for Time to Contact from Visual Information - A Simple Approach That Requires No Recognition of Objects –“International Conference, IEEE 2014, Bali, Indonesia, 5-10th December, Robotics and Biomimetics, 2014, pp. 469-474.

A Round-Trip Engineering Method for Activity Diagrams and Source Code

Keinosuke Matsumoto, Ryo Uenishi, and Naoki Mori
Department of Computer Science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
Sakai, Osaka, Japan
email: {matsu, uenishi, mori}@cs.osakafu-u.ac.jp

***Abstract*—In the field of software development, many implementation methods appear one after another. It is necessary for them to be flexibly introduced into software. Model driven development is regarded as one of the most flexible development methods. It expects to generate source code from the models. However, the models and the source code generated from them will become out of sync if the code is changed. In order to solve this problem, round-trip engineering (RTE) has been proposed. RTE has a feature that keeps the models synchronized with the source code. There are some tools providing us with the RTE, but almost all of them are applicable only for static diagrams. This research adapts the RTE directly to activity diagrams as one of dynamic diagrams, and proposes a method to realize the RTE for activity diagrams and source code. A success transformation rate of the models and source code has been confirmed. As a result, it could be verified that the round-trip engineering between activity diagrams and source code is successful.**

Keywords-model; round-trip engineering; activity diagram; model driven development; UML.

I. INTRODUCTION

Model driven architecture (MDA) [1][2] draws attention as a technique that can flexibly deal with changes of business logics or implementation technologies in the field of system development. Its core data are models that serve as design diagrams of software. It includes a transformation to various kinds of models and an automatic source code generation from the models [3][4][5].

Development standardization is advanced as model driven architecture by Object Management Group (OMG). However, the models and the source code generated from them will become out of sync if the code is changed. In order to solve this problem, round-trip engineering (RTE) [6][7][8][9] has been proposed. RTE has a feature that keeps the models synchronized with the source code. Therefore, it is possible to keep them consistent. There are some tools providing the RTE, but almost all of them are applicable only for static diagrams such as class diagrams, component diagrams. Therefore, it is necessary to adapt the RTE to dynamic diagrams.

This research adapts the RTE to activity diagrams as one of the dynamic diagrams, and proposes a method to realize the RTE for activity diagrams and source code [10][11]. Activity diagrams are defined in Unified Modeling Language (UML), and describe flows of activities. They can also express processes hierarchically and are used widely from upper to lower processes of software development. Figure 1 shows a basic concept of the proposed method. In transforming activity diagrams to source code, the proposed method analyzes XML metadata interchange (XMI) [12] of the activity entities. XML is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. XMI is a standard for exchanging metadata information. Conversely, in transforming source code to activity diagrams, the proposed method analyzes the abstract syntax tree (AST) [13] of the source code. In mutual transformation of them, an intermediate representation is used. It has hierarchical structure, and corresponds to both activity diagrams and source code. For this reason, you can easily transform between XMI and AST. Describing conditional branches and loop statements, activity diagrams use the same elements. They cannot be transformed to source code as they are. Therefore, a method for analyzing them and mutual transforming is developed for distinguishing the conditional branches and loop statements. A success transformation rate of the models and source code has been confirmed. As a result, it could be verified that the validity of the proposed method.

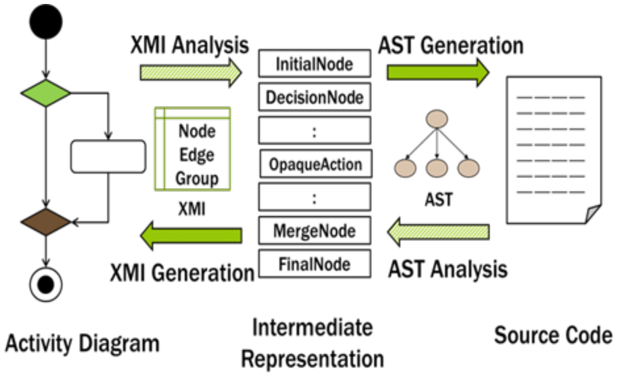


Figure 1. Schematic diagram of the proposed method.

The contents of this paper are shown below: Section II describes related work. Section III explains the proposed method of this research. Section IV shows the results of application experiments in order to confirm the validity of the proposed method. Finally, Section V describes conclusion and future work.

II. Related Work

This study uses related work called AST and RTE.

A. *Abstract Syntax Tree*

AST that belongs to Eclipse AST implementation is a directed tree showing the syntactic analysis results of source code. It is also used in order to create byte code from the source code as internal expression of a compiler or an interpreter. AST provides us with ASTParser class which changes source code into AST. There are many kinds of nodes defined by AST. An AST node can be searched by using ASTVisitor class corresponding to one of design patterns [14]. The visitor design pattern is a way of separating an algorithm from an object structure on which it operates. A practical result of this separation is the ability to add new operations to existing object structures without modifying those structures. An example of AST is shown in Figure 2. Detailed analysis can be carried out by changing AST levels.

B. *Round-Trip Engineering*

RTE refines intermediate results by editing requirement definitions, design plans, and source code alternately. Generally, if either models or code is changed, the RTE automatically reflects the change on the other side. RTE has a feature that keeps the models synchronized with the source code. The outline of RTE is shown in Figure 3.

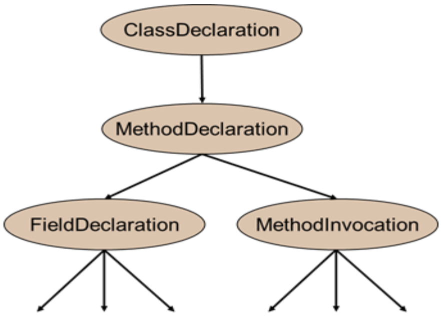


Figure 2. An example of AST.

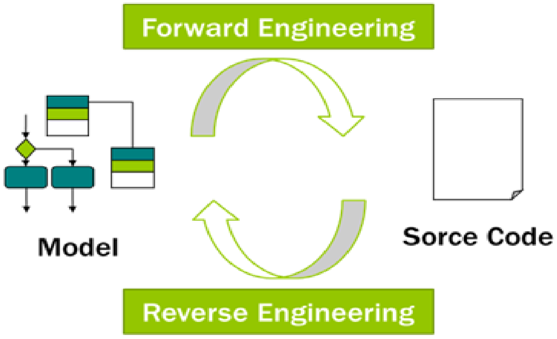


Figure 3. Outline of RTE.

Some tools, like UML Lab [15] and Fujaba [16][17], are proposed to maintain consistency of models and source code. In these tools, a template for generating source code is described by a template description language. Automatic generation of source code can be carried out from models by using the template. It enables to refactor source code and static diagrams, such as class diagrams and component diagrams, synchronously. It also does code generation and reverse engineering in real time. However, it does not deal with dynamic diagrams like activity diagrams which can describe the behavior of a system. Although Fujaba considers activity diagrams, the tool does not address them in a direct way. On the other hand, our approach deals directly with the activity diagrams.

III. Proposed Method

This section proposes a transformation method from activity diagrams to source code and from source code to activity diagrams. Activity diagrams mainly describe the behaviors of a system using nodes and edges. A content of action is described in a node. The flow of a series of actions is expressed by connecting nodes by edges. An activity diagram is described for each method in class diagrams in the proposed method.

A. *Transformation from Activity Diagram to Source Code*

A concrete transformation flow from activity diagrams to source code is as follows:

1) XMI Analysis of Activity Diagram: An activity diagram is expressed in XMI form as an UML file. It begins with a start node and ends with a final node, following some nodes or groups through edges. Nodes have information on actions or controls of the activity diagram. Edges have information on control flows as some attributes and subelements. Group is a tag that has nodes and edges of a subactivity as subelements. Each tag is given an id for discriminating from other tags. Table I shows nodes used by an activity diagram.

2) Transformation from XMI to Intermediate Representation: Node and edge tags have a transition starting id and targeting id respectively. Using these ids, you can extract the flow of actions of an activity diagram as a sequence of ids. It can be transformed to an intermediate representation by replacing ids with corresponding nodes

TABLE I. Nodes used by an activity diagram.

Tag	Node
Node tag	ActivityInitialNode
	ActivityFinalNode
	CallBehaviorAction
	CallOperationAction
	DecisionNode
	LoopNode
	MergeNode
	OpaqueAction
Group tag	StructuredActivityNode
Edge tag	ControlFlow

extracted from XMI analysis. The intermediate representation is a sequence of nodes as the flow of actions. The reason for introducing the intermediate representation is because it makes it easy to transform both XMI and source code into one another. Figure 4 shows a metamodel of intermediate representation, and Figure 5 shows the image of this transformation.

3) *Transformation from Intermediate Representation to AST:* Analyzing the flow of the actions of an intermediate representation, you can transform it into AST. The inter mediate representation is analyzed in order from the beginning. According to corresponding nodes, it is necessary to extract information, such as a branch and loop, from the representation structure. For example, a branch has a structure embraced by Decision node and Merge node, but a loop has a structure embraced by Decision nodes. In order to distinguish such structures, a stack which stores ids of Decision nodes is created. If a Decision node comes out, the id is pushed to the stack at once. It is a branch if a Merge node comes out before a Decision node comes out next. If a Decision node comes out and its id is the same id pop from the stack, then it is a loop. Otherwise, a new Decision node comes out and its id is stacked. Figure 6 shows this transformation.

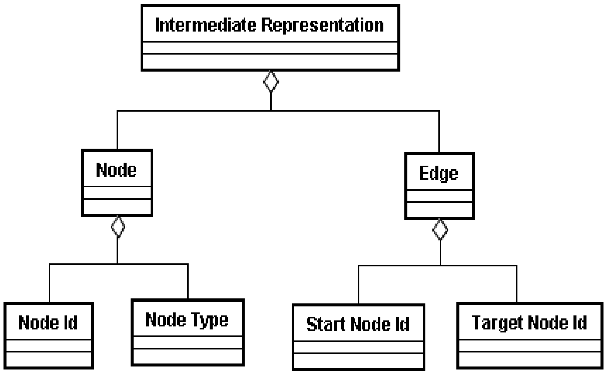


Figure 4. Metamodel of intermediate representaion.

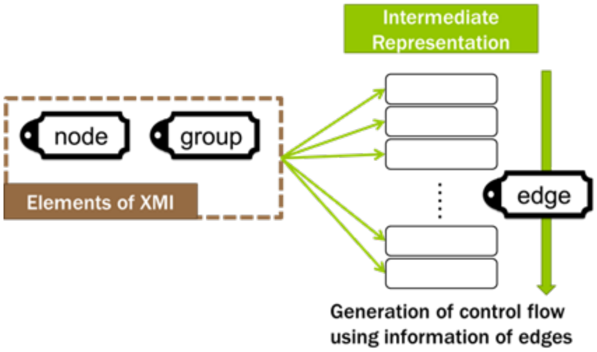

Intermediate Representation
node
group
Elements of XMI
edge
Generation of control flow using information of edges

Figure 5. From XMI to intermediate representation.

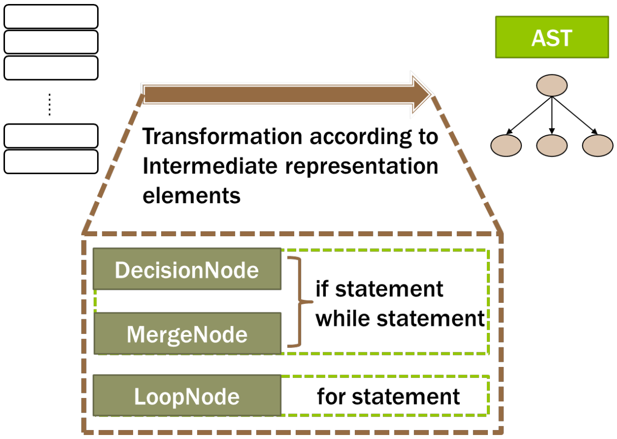


Figure 6. From intermediate representation to AST.

4) *Transformation from AST to Source Code:* Target source skeleton code is transformed from class diagrams by using Acceleo templates for classes. Acceleo [18] is the Eclisp Foundation’s open-source code generator which provides us with templates for skeleton code. Transformed activity diagrams and classes of a target source skeleton code are expressed by AST. A method whose name is identical with that of an activity diagram can be searched by using ASTVisitor class. The method code transformed from AST of the activity diagram is added to the method body to which corresponds in the target source skeleton code for every activity diagram.

B. Transformation from Source Code to Activity Diagram

A concrete flow of transforming from source code to activity diagrams is as follows:

1) *AST Analysis of Source Code:* ASTParser class transforms source code into AST, and ASTVisitor class searches AST nodes to deal with. These are defined as AST library. The structure of source code is analyzed by using these classes.

2) *Transformation from AST to Intermediate Representation:* Required information is extracted by analyzing AST. Whenever an AST node is searched, the information on the AST node is saved in detail. Required AST nodes are DeclarationStatement node (like variables, call of methods), IfStatement node, WhileStatement node, ForStatement, and so on. The flow of the processing is almost the same as that of the transformation from XMI of an activity diagram to intermediate representation. Figure 7 shows this transformation.

3) *Transformation from Intermediate Representation to XMI:* A sequence of ids could be extracted from nodes, groups, and edges in the transformation from activity diagrams to source code. If this transformation is carried on in reverse, nodes, groups, and edges are generated by analyzing the flow of actions. Specifically, nodes or groups are generated for each action of the intermediate representation. They are transformed to XML according to

the kind of actions. Simultaneously, edges which connect between nodes or groups are generated. A transition starting id and targeting id are generable from the sequence of intermediate representation. Generating Decision or Merge nodes expressing branches or loops, a stack which is similar to that of the transformation from activity diagrams to intermediate representation is used.

4) Adding XMI to Activity Diagram: Generated nodes, groups, and edges are added to XMI file of an activity diagram. In case of adding, you refer to the activity diagram in the package where the source code is allocated. If the diagram already exists, adding is performed after deleting the contents of the existing file. Otherwise, adding is performed after generating a new diagram.

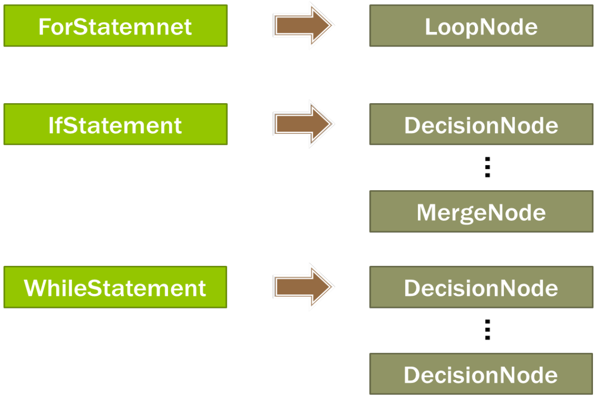


Figure 7. From AST to intermediate representation.

IV. Application Experiments

The proposed method is applied to a hunter game [19] to confirm the effectiveness of the proposed method. We have both activity diagrams and source code of the hunter game. The number of AST nodes of original hunter game is 4971. Mutual transformations of the activity diagrams and the source code are carried out by the proposed method. As a result, Figure 8 describes comparison results of the number of AST nodes. Tables II and III show the comparison of the number of XMI and AST nodes respectively.

The transformation rate is computed by comparing the number of XMI nodes of activity diagrams. The objects to compare are handwritten activity diagrams and the activity diagrams automatically generated from the source code.

TABLE II. Comparison of the number of XMI nodes.

XMI node	Automatic	Original	Difference
group	47	47	0
guard	155	159	-4
edge	1137	1142	-5
node	1367	1369	-2

TABLE III. Comparison of the number of AST nodes.

AST node	Automatic	Original
SwitchCase	0	4
SwitchStatement	0	1
CatchClause	0	8
TryStatement	0	8
VariableDeclaration	69	77
Block	364	315

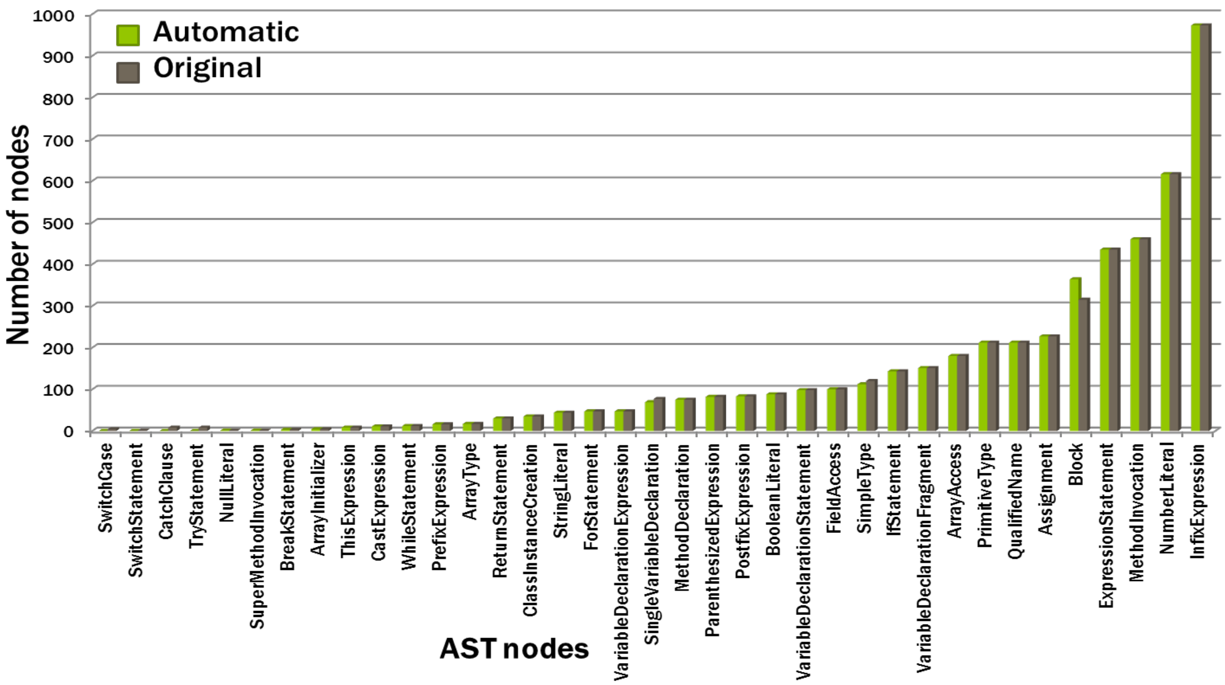
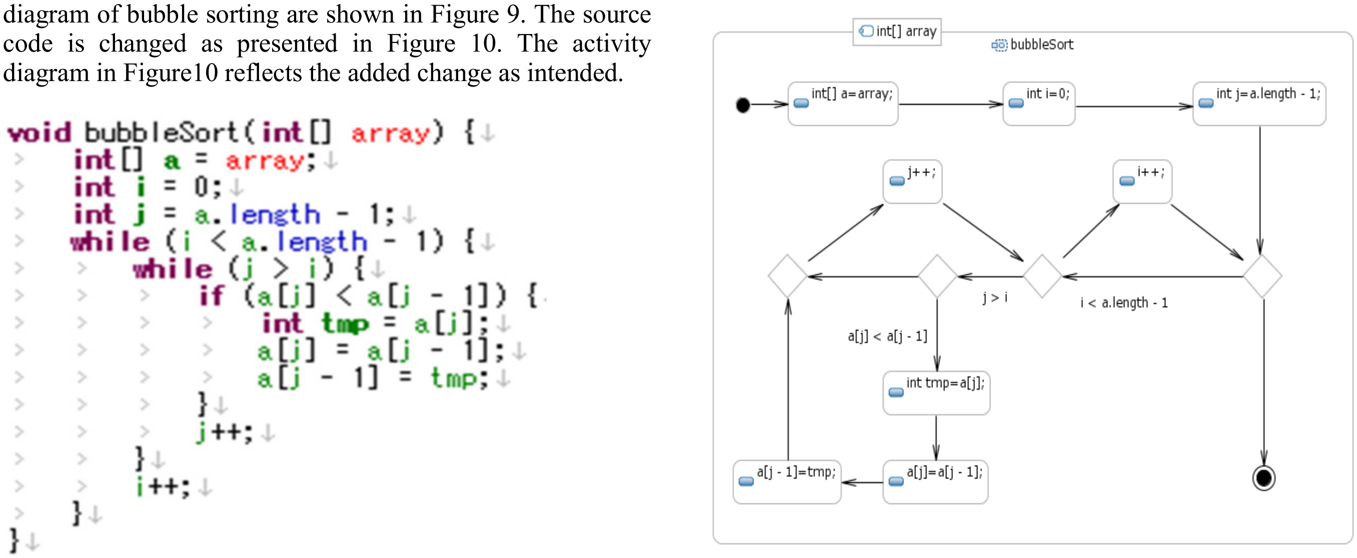


Figure 8. Comparison results of AST nodes.

The transformation rate is 99.6% (= generated XMI nodes * 100 / original XMI nodes). XMI nodes which are not transformed are shown in Table II. There are three kinds of nodes: guard, edge, and node. A switch statement cannot be described in an activity diagram, but the same processing can be described by using if statements. Guard nodes decreases in the same number of switch statements in generated activity diagrams. The number of edges is also decreasing in connection with it.

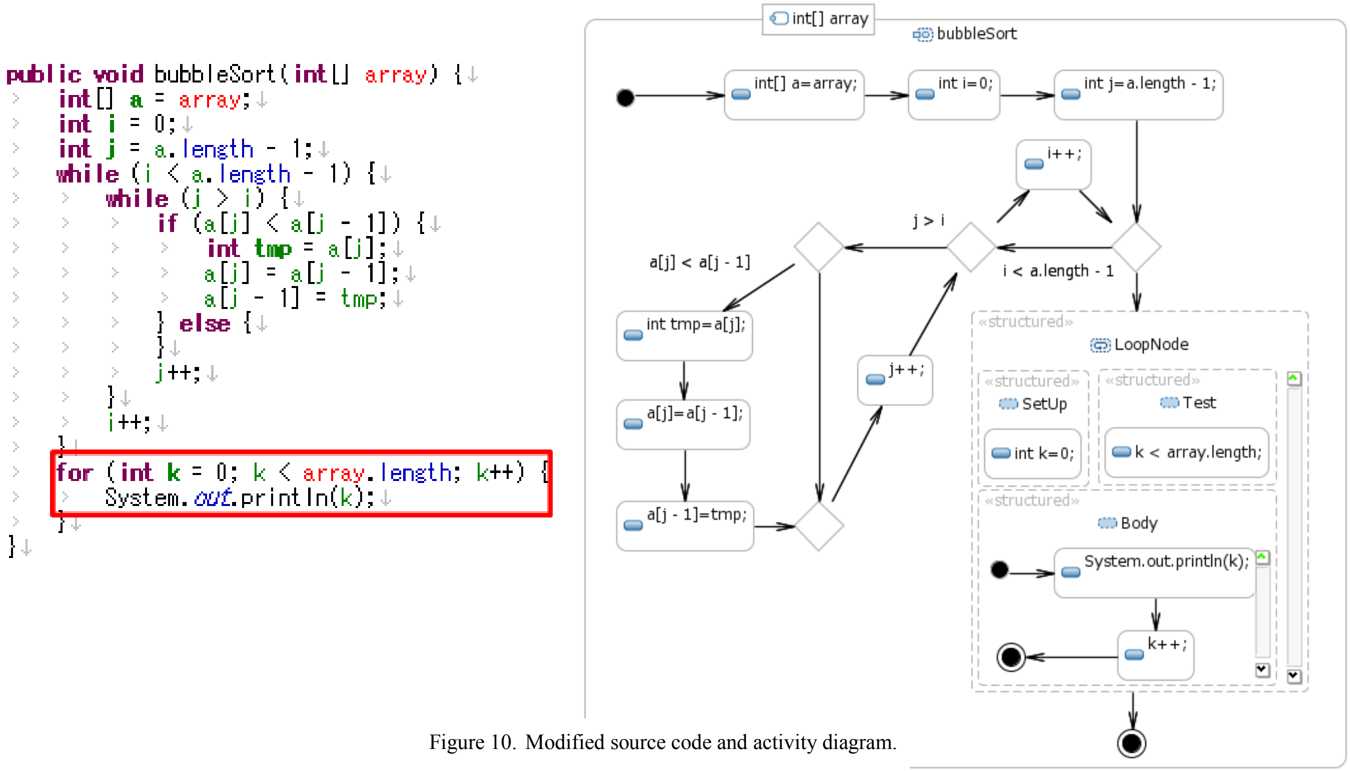
After adding change to source code, an activity diagram is generated from the changed source code. It is verified whether the generated activity diagram reflects the added change. For example, original source code and activity diagram of bubble sorting are shown in Figure 9. The source code is changed as presented in Figure 10. The activity diagram in Figure10 reflects the added change as intended.

A reverse transformation is investigated by generating activity diagrams from handwritten source code and transforming from these activity diagrams to source code. The objects to compare are handwritten source code and the automatic generated source code. The transformation rate is 99.8%. Except for switch statements and the positions of block, they are almost similar. It is verified that the generated source code is functionally equivalent to the handwritten source code. The transformation rates for forward and reverse transformation are not 100% because there is no standard expression to describe switch and try-catch statements in an activity diagram. They are not transformed by the proposed method as shown in Table III.



```
void bubbleSort(int[] array) {
    int[] a = array;
    int i = 0;
    int j = a.length - 1;
    while (i < a.length - 1) {
        while (j > i) {
            if (a[j] < a[j - 1]) {
                int tmp = a[j];
                a[j] = a[j - 1];
                a[j - 1] = tmp;
            }
            j++;
        }
        i++;
    }
}
```

Figure 9. Original source code and activity diagram.



```
public void bubbleSort(int[] array) {
    int[] a = array;
    int i = 0;
    int j = a.length - 1;
    while (i < a.length - 1) {
        while (j > i) {
            if (a[j] < a[j - 1]) {
                int tmp = a[j];
                a[j] = a[j - 1];
                a[j - 1] = tmp;
            } else {
            }
            j++;
        }
        i++;
    }
    for (int k = 0; k < array.length; k++) {
        System.out.println(k);
    }
}
```

Figure 10. Modified source code and activity diagram.

V. CONCLUSION

This paper has pointed out a problem in model driven development and proposed a method of applying round-trip engineering to activity diagrams in order to solve the problem. The effectiveness of the proposed method is verified by the application experiments for the source code of a hunter game. Consequently, it has confirmed that the round-trip engineering between activity diagrams and source code is successful. The characteristics of the activity diagrams accepted by this approach are as follows: They consist of actions of the same granularity, not so many multilayered group nodes.

Since activity diagrams cannot yet deal with switch and try-catch statements, defining of these description methods and increasing convertible elements are important as future work.

ACKNOWLEDGMENT

This work was supported in part by JSPS KAKENHI Grant Number 24560501.

REFERENCES

[1] S. J. Mellor, K. Scott, A. Uhl, and D. Wiese, MDA Distilled: Principle of Model Driven Architecture, Addison Wesley Longman Publishing Co., Inc. Redwood City, CA, 2004.

[2] S. Beydeda, M. Book, and V. Gruhn, Model-Driven Software Development, Springer Berlin Heidelberg, 2005.

[3] A. Uhl, "Model-Driven Development in the Enterprise," IEEE Software, January/February 2008, pp. 46-49.

[4] R. F. Paige and D. Varró, "Lessons Learned from Building Model-Driven Development Tools," Software System Model, Vol. 11, 2012, pp.527-539.

[5] N. Condori-Fernández, J. I. Panach, A. I. Baars, and T. Vos, Ó. Pastor, "An Empirical Approach for Evaluating the Usability of Model-Driven Tools," Science of Computer Programming, Vol. 78, No. 11, 2013, pp. 2245–2258.

[6] N. Medvidovic, A. Egyed, and D. S. Rosenblum, "Round-Trip Software Engineering Using UML: From Architecture to Design and Back," Proc. of the 2nd Workshop on Object Oriented Reengineering, 1999, pp.1-8.

[7] U. Aßmann, "Automatic Roundtrip Engineering," Electronic Notes in Theoretical Computer Science, vol. 82, 2003, pp. 33-41.

[8] A. Henriksson and H. Larsson, "A Definition of Round-Trip Engineering," Technical Report, University of Linköping, Sweden, 2003.

[9] M. Antkiewicz and K. Czarnecki, "Framework-specific modeling languages with round-trip engineering." in Model Driven Engineering Languages and Systems, Springer Berlin Heidelberg, 2006, pp. 692-706.

[10] A. K. Bhattacharjee and R. K. Shyamasundar, "Activity Diagrams : A Formal Framework to Model Business Processes and Code Generation," Journal of Object Technology, Vol. 8, No. 1, January-February 2009, pp. 189-220 .

[11] Pakinam N. Boghdady, Nagwa L. Badr, Mohamed Hashem, and Mohamed F. Tolba "A Proposed Test Case Generation Technique Based on Activity Diagrams," International Journal of Engineering & Technology IJET-IJENS Vol. 11 No. 3, 2011, pp. 35-52.

[12] XML metadata interchange. XMI: [Online]. Available from: http://www.omg.org/spec/XMI/ 2015.3.18.

[13] I. Neamtiu, J. S. Foster, and M. Hicks., "Understanding Source Code Evolution Using Abstract Syntax Tree Matching," ACM SIGSOFT Software Engineering Notes. Vol. 30. No. 4. ACM, 2005, pp. 1-5.

[14] E. Gamma, R. Helm, R. Johson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

[15] Unified Modeling Language Lab. UML Lab: [Online]. Available from: http://www.uml-lab.com/en/uml-lab/ 2015.3.18.

[16] U. A. Nickel, J. Niere, J. P. Wadsack, and A. Zündorf, "Roundtrip Engineering with Fujaba," Proc. of the 2nd Workshop on Software-Reengineering (WSR), Bad Honnef, 2000, pp. 1-4.

[17] L. Geiger and A. Zundorf, "Tool Modeling with Fujaba," Electronic Notes in Theoretical Computer Science, vol. 148, 2006, pp. 173-186.

[18] Acceleo: [Online]. Available from: http://www.eclipse.org/acceleo/ 2015.3.18.

[19] M. Benda, V. Jagannathan, and R. Dodhiawalla, "On Optimal Cooperation of Knowledge Sources," Technical Report, BCS-G 2010-28, Boeing AI Center, 1985.

Platform for Autonomous Service Composition

Krasimir Baylov, Dessislava Petrova-Antonova, Aleksandar Dimov
Department of Software Engineering
University of Sofia “St. Kliment Ohridski”
Sofia, Bulgaria
e-mail: krasimirb@uni-sofia.bg, d.petrova@fmi.uni-sofia.bg, aldi@fmi.uni-sofia.bg

Abstract — The increasing complexity of business processes requires improved methods for composition of web services. In order to fulfill this, it is difficult for administrators to keep up with the growing demand and the enormous amount of customizations required by the users. A possible solution that will help in this situation is to develop methods and technologies that support autonomous compositions of web services. Such compositions should adapt dynamically to changes in the requirements or the environment. This paper describes a platform which implements such solution, based on Quality of Service (QoS). The platform prototype, which is presented here, is able to monitor web service QoS and determine whether the service composition fulfils the overall quality required by the end users.

Keywords - Quality of Service; Web services; SOA; Dynamic web services composition

I. Introduction

Service Oriented Architectures (SOA) play an important role in enabling integration of business with IT [25]. Services are a key concept in SOA and they represent reusable entities that should minimize the development effort and provide means for information exchange for both service consumers and providers. On the other hand, the complexity of business problems is increasing and to solve them, users could employ a number of services into a composition to execute a business process. However, as business gets more and more flexible today, consumers require additional functionality and customizations towards the services they use. In other words, a static composition is not capable to fulfill all user requirements in a long term perspective. This makes service providers search for ways to deal with the increasing number of service demand while at the same time providing personalized Service Level Agreement (SLA) management [26].

A solution to this problem is to provide compositions that are autonomic and are capable to adapt to changes in user requirements or the environment. Such compositions can adapt according to some measurable rules. Let us consider that, for a composition, one should choose a service out of a set of services that share similar functionality. To solve that, it is possible to choose a service that offers the best Quality of Service (QoS). Moreover, it would be better if the composition is not static but changes dynamically according to changes in the QoS of services (based on changes in workload, number of requests, etc.) or in user requirements. For example, if more users send requests to a service, its response time may raise to an undesirable level, and then another service should be found and integrated into the composition. The goal is to make this with minimal human intervention and implement the change dynamically and transparently for the user.

This paper presents a platform for building autonomous web service compositions based on QoS. The platform provides means for gathering data for evaluation of service QoS characteristics (like performance, availability, reliability, cost, etc.). Such means include:

- An extended service registry, which is used as a repository for collection of service QoS data and enables easy web-services search and selection
- A model to calculate service QoS, according to the data in the registry
- An algorithm to find and select the services that will best meet the agreed SLA of the composition
- Automatically and transparently integrate selected web services into a working composition

A key aspect of the proposed platform is that service compositions are determined and updated dynamically at runtime. This frees administrators and developers from implementing any QoS related changes.

The rest of this paper is organized as follows: Section 2 makes an overview of the related work; Section 3 presents the model that we use for evaluating the quality of web service compositions and determining the best composition; Section 4 introduces the design and implementation of our platform for autonomous web service composition. Section 5 presents a simple case-study to illustrate usage of the platform and validate it, and finally, Section 6 concludes the paper and states directions for further research.

II. Related Work

Quality attributes are very important in terms of design and reasoning about of software systems. They are regarded as key concerns in software architecture design [3] and selection of relevant web services [9]. Many researchers have also managed to solve the problem with formal definition and management of software quality in general. For example, there exist a lot of theoretical models for evaluation of reliability [6][24], performance [2][7], complexity, etc. However, such models tend to be relevant only at theoretical level as they are either quite general and have some unrealistic assumptions that make them inapplicable in practice or too complex to be applied in a broad range of domain areas.

The work that relates to ours may be split into two main directions: first one is related to models for calculation of software QoS and second one – to methods for dynamic and autonomous web service composition. Many research efforts combine the two directions and use the overall QoS of a service composition to determine whether it should change or not [11][13]. There are a lot of models available that consider the quality characteristics and based on them provide the best service composition.

Liu et al. [22] proposed of QoS model that is open and extensible. They provide an implementation of a QoS registry that stores the web services QoS data and allows consumers to search against it. A key point here is that such data is obtained through user feedback, i.e., consumers that use the services rate them and provide their feedback to the QoS registry.

Ran [18] proposed a model for web service discovery based on QoS. They argue that current web service registries limit service discovery to functional requirements only and non-functional properties should be paid more and more attention. They extend the current web services registration and discovery model by introducing a new role – *Web Service QoS Certifier*. The concept of a certifier is also covered in [12]. The certifier is responsible to certify/verify the claimed non-functional properties of the web service providers.

The DYSCO platform [15] provides a complex solution for dealing with dynamic web service composition. The platform allows automatic generation of executable business processes and SLA for each web service. It also provides mechanisms for monitoring the used web services and updating the business process when SLA deviations are discovered.

AgFlow [10] is a middleware platform that allows quality-driven dynamic web services composition. The platform provides a multidimensional QoS model that is responsible for capturing the non-functional properties of the web services. This work introduces two approaches for selecting web services – *local optimization* and *global planning*. An adaptive execution engine is responsible for the runtime adaptation of the web services composition. It replans the execution any time when any of the services is unavailable or the quality properties exceed predefined thresholds.

The web service composition algorithm proposed by Lu et al. [19] is based on seven QoS properties – running cost, runtime, success ratio, usability, trustworthiness, degree of security and degree of semantic correlation. A limitation of the algorithm is that only semantic (immeasurable) QoS properties are considered. In addition, it is not clear how the QoS properties are assessed. In contrast, Yu et al. [20] rely on measurable QoS properties and especially on latency, execution cost, availability and accuracy. The advantage of the proposed solution is that it is applicable to data intensive web service compositions. It combines the tabu search and the genetic programming techniques. The last one is applied also in the web service composition approach presented in [1].

An approach for self-healing web service composition is introduced by Aziz et al. [14]. It repairs the web service composition when some of its components violate the QoS constraints. The headers of the SOAP messages are extended in order to provide information about QoS properties. The approach includes three main phases: monitoring, diagnosis and repairing. When QoS degradation is detected during diagnosis phase, a repairing procedure is started. As a result the failed web service is replaced with another one obtained from the UDDI registry. A possible drawback of the approach is that it relies on SOAP as communication protocol and is not clear how it could be applied when the composition includes REST web services.

The web service composition system presented by Brahmi and Gammoudi [23] is based on cooperative agents. The agents are organized as a social network and cooperate to find the optimal composition with respect of QoS. The approach proposed by Xia and Yang [21] is focused on QoS optimization and redundancy removal. An advantage of the approach is that it removes most of the redundant web services minimizing total execution cost of the composition. Unfortunately, the QoS optimization is based only on two QoS properties – response time and throughput.

Birgit and Marchand-Maillet [5] solved the web service composition problem partially by providing an algorithm for QoS-aware selection. The algorithm uses a rank aggregation instead of direct measures of QoS values. Its core includes so called abstract voter that sorts the web services according to a particular QoS property, named QoS factor. In [8], the web service composition problem is formalized as problem of traversing a Petri Net. The estimation of composition's quality is performed through utility function that aggregates the functional, QoS and transactional properties of the web services.

Currently, there is no universal approach for autonomous management of dynamic web service composition based on QoS. In this work, we propose a platform that deals runtime with QoS monitoring, adaptation and discovery of web services, in order to determine the best possible composition. Another advantage of the approach presented here is that it is compatible with the Business Process Execution Language (BPEL) standard and is capable of implementing an executable composition.

III. A MODEL FOR AUTONOMOUS WEB SERVICE COMPOSITION

In this section, we present the model we use for autonomous web service composition. It includes analyzing the quality data for each eligible web service and determining the best composition that matches a predefined set of quality requirements. The presented model is based on [16] and [17], but adds the following additional features to achieve the goal:

- Introduces the concept of a web service category as an abstract entity that may refer to multiple web services, providing the same functionality and interface.

- Uses weight (i.e., priorities) of quality attributes to determine the weight of actual user requirements.
- Analyzes all service compositions that may be integrated in order to find the one that best matches the user defined business process.

Moreover, the model introduces the concept of a *web service category*. Each category is defined by common functionality and an interface and may be associated with multiple web services. Therefore, the presented model requires that users define their business processes by specifying the web service categories rather than the concrete web service implementation. In other words, each business process is considered as a composition of multiple web service categories. Concrete web services are assigned after the model is applied and the best composition is known.

Currently, to our best knowledge, there is no unified standard for managing web service categories. For the purpose of this research, we have used a central service registry for discovering web service categories and associated web service implementations for each of them.

The first step in our model is determining the set of web service categories that build our business process. In this case, we are not interested in the sequence of the particular web service invocations but need to know the set of different web service categories like

$$BP = \{WSC_1, WSC_2, \dots, WSC_n\} \quad (1)$$

WSC$_n$ refers a single web service category that is used in the business process definition.

Additionally, we need to know the specific web service implementations associated with each category

$$WSC_i = \{WS_{i_1}, WS_{i_2}, \dots, WS_{i_m}\} \quad (2)$$

In this case, WSC denotes a single service category and WS denotes a particular web service implementation. A single web service category WSC may include multiple web service implementations WS.

We also need the set of quality requirements R and their associated weights C.

$$R = \{r_1, r_2, \dots, r_l\} \quad (3)$$
$$C = \{c_1, c_2, \dots, c_l\} \quad (4)$$

Requirements and weights are set by the business process designer. They should reflect the end user needs. Requirements represent the quality characteristics under consideration like performance, availability, throughput, etc.

Weights are measured by relative values ranging between 0 and 1. Naturally, the sum of all weights should be equal to 1. l represents the number of quality attributes under consideration (performance, availability, etc.). Note that the requirements and the weights are paired. For each requirement r_i, there is an associated weight c_i.

Then, for each web service, the relevant quality data should be presented in a matrix. Each row represents an execution of the web service and each column represents a quality attribute $\{r_1, r_2, \dots, r_l\}$.

$$R_{WS_{ij}} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1l} \\ x_{21} & x_{22} & \cdots & x_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kl} \end{bmatrix}, i = 1 \div n, j = 1 \div m \quad (5)$$

In this matrix, k specifies the number of web service calls. Each row in this matrix represents the different quality characteristics for the related call like response time, throughput, available or not, etc. The data in this matrix is dynamic. It changes as new web service calls are invoked and quality data is updated. Note that we may not need to analyze the entire set of web service calls but only a subset of them. For example, in many cases it may be more practical to analyze only the last number of calls. This number may be updated dynamically based on the platform that uses the presented model.

The next step in this approach is to calculate the quality attribute values and normalize them so that they could be easily compared. For this purpose we need the average, minimum and maximum values for each quality characteristic from the $R_{WS_{ij}}$matrix. This means that we need separately process each column in the matrix. The average value is the sum of all x values for a quality characteristic divided by their number. The min and max values represent the lowest and highest values respectively. Therefore, for each column z ($z = 1 \div l$), we calculate the normalized value for the quality characteristics.

$$P_{zWS_{ij}normalized} = \frac{<average>}{|<max> - <min>|}, z = 1 \div l \quad (6)$$

Once we have the normalized quality characteristics values, we should sum them in order to get a numeric representation of the web service quality.

$$R_{WS_{ij}} = \frac{\sum_{z=1}^{l} P_{zWS_{ij}normalized} \cdot c_z}{l} \quad (7)$$

R_{WS} represents the normalized quality value of the j-th web service from category *i*. It is important to consider the weight/priority of each quality attribute. Therefore, the value representing the overall quality for a single web service would be the sum of the normalized quality values for each quality characteristic multiplied by the relevant weight factor.

By now, we should have a numeric representation of the quality of each web service. Next we analyze all combinations of web services in the composition to find the best one. For each possible composition we calculate the related quality by summing the quality values for each service that builds it.

$$Q_{BP} = \sum R_{WS_{ij}} \quad (8)$$

Once the quality value for each web service composition Q$_{BP}$ is calculated, we analyze them and select the one that has achieved the highest score. As a result, a single web service composition is selected. This is the one that best meets the user requirements according to the model.

This model allows us to find the best web service composition based on a predefined set of quality requirements and their associated weights. It could easily be applied to any number of web services and extended to support various types of quality attributes. This model is applied in the implementation of our autonomous service composition platform. The technical implementation of the platform is presented in Section 4.

IV. QOS BASED PLATFORM FOR AUTONOMOUS SERVICE COMPOSITION

In this section, we present the design and implementation of our platform. It is based on the model described in the previous section and follows the architecture presented on Figure 1. The model for determining the best service composition is implemented in the *BPEL Extension* component. The current platform allows runtime updates to the deployed service compositions with no human supervision.

Our platform also provides an extended web service registry that allows consumers to search for the services they need and also inquire information for their QoS characteristics. Finally, all these data are processed by a BPEL extension tool which is part of our previous work and it allows dynamic binding of the selected web services in the defined composition [4]. The platform consists of the following components:

1. **BPEL Extension** – extension deployed on business process server allowing to perform the runtime composition of web services and adjust to the quality requirements of each user.
2. **Extended Service Registry** – a standard service registry with a DB extension for persisting quality attributes data for the web services. Access to this data is exposed as part of the registry interface allowing service consumers to use it for their composition analysis.
3. **Web Service Interceptor** – tool that is able to intercept any web service call and collect the needed quality attributes data. This data is then stored in the extended service registry and made available of other service consumers.

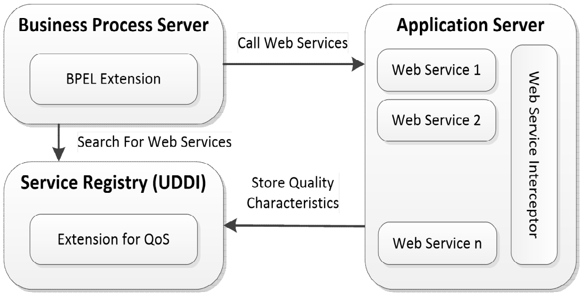


Figure 1. Architecture of the platform for dynamic web service

In the next subsections, we provide a detailed description for each of the platform components.

A. BPEL Extension

Our BPEL extension allows updating a BPEL process at runtime. It is developed according to the BPEL extension specification and can be deployed and plugged in any BPEL compliant server. In this work, the WSO2 BPS server is used to test the extension.

B. Extended Service Registry

The extended service registry provides the standard UDDI (Universal Description, Discovery and Integration) interface. Already existing *Apache jUDDI v.3.0.4* registry is used for this purpose. All web services that the platform can work with are registered there.

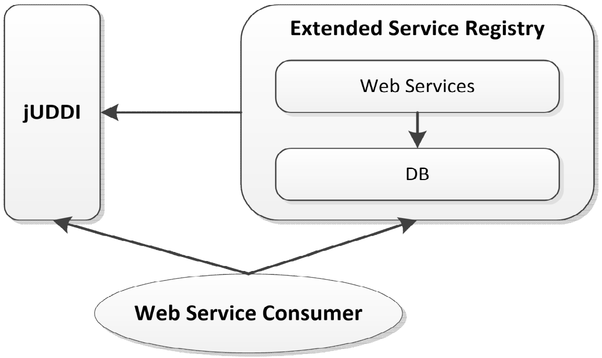


Figure 2. Extended service registry design approach

As stated in the name of the component it provides extended functionality. We have deployed a database that stores the quality characteristics for each web service invocation. This data is stored in raw format so that it can be used with various models. To make this data accessible we have developed Apache jUDDI-like services (https://juddi.apache.org/), so that consumers could obtain the quality data they need for building their service composition. Those web services are exposed as SOAP services.

In order to make the extension as loosely coupled to the UDDI registry we have implemented it as a separate tool that end users could integrate with. Figure 2 represents our design approach.

This approach allows service consumers to use the UDDI registry in a standard way and only those who are interested in the QoS data could trigger the relevant queries against the extension. In addition, our extension is aware of the service categories.

A key point here is that we try to avoid the concept of using a web service QoS certifier. We would not let service providers publish any QoS data for their web services. Rather, we would expect every provider that is interested in providing such data to install the so-called web service interceptors that we provide. They will store the relevant data in the extended registry. This way the interceptors (presented in the next section) act like a certified web service QoS data provider.

C. Web Service Interceptor

The web service interceptor is a module, responsible for gathering quality related data for each web service invocation. This module is deployed on each server that hosts the implemented web services. Because there are multiple technologies for implementing and exposing web services we have limited ourselves to using the Apache Axis 2 framework. It provides mechanisms for extensibility and we could easily integrate our custom logic. What's more the Apache framework design includes mechanisms for developing custom handlers for the supported web services.

We take advantage of this functionality and we have developed custom handlers that intercept the web service invocations. There are two types of handlers – *message flow handlers* and *error flow handlers*. The *message flow handlers* process the standard web service invocation while the *error flow* ones are activated when the web service fails. Figure 3 shows how the interceptors fit into the process of a web service invocation.

For each web service invocation we get the following data – S*OAP message size*, *processing time* and identification data like operation correlation ID, IP addresses, etc. This data is then stored in our database and exposed for calculation of quality data.

One of the major design goals for web services interceptor module is modularity. Therefore, it is implemented in an easy to configure and customize way. The interceptor module itself is packaged as a ".mar" (module archive) file. This file is deployed on the servlet container by creating a folder named "modules" in the "webapps/axis2/WEB-INF" directory.

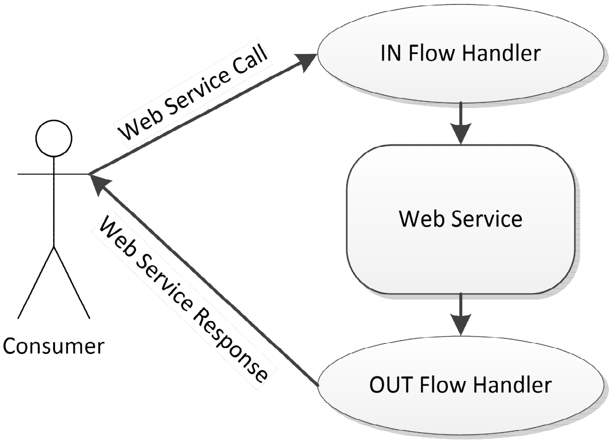


Figure 3. Intercepting web services invocation

V. Experiments

In order to show the benefit of the Autonomous Service Composition Platform, this section presents experiments that show how it performs in selection of the best (by QoS) web service. The current implementation is still a prototype and additional validation will be made when it matures. The experiment focuses on two quality characteristics – performance and availability. We have set the weight for each for the quality characteristics to 0.7 for performance and 0.3 for availability. For the purpose of this experiment, we have defined three *web service categories*, each representing a mathematical operation – *Multiply*, *Power* and *Add*. For each category we have developed a set of three web services with the same functionality and interface but simulating different quality characteristics – *standard*, *slow* and *randomly available*.

To make the experiment we created a business process that uses the three web service categories. Each of them is called one after another. In this case, we are not interested in the final result of the calculation but we pay close attention to the quality characteristics of the executed business process. Figure 4 represents the business process we use in our experiments.

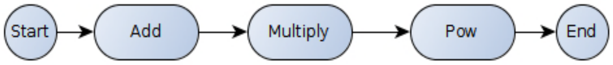


Figure 4. Experimental business process

From a model perspective, our composition can be presented in this way

$$BP = \{WS_{Multiply}, WS_{Power}, WS_{Add}\} \qquad (9)$$

After a series of service invocations we collected data regarding the quality characteristics of each web service. Table 1 presents the obtained average values.

TABLE I. Aggregated QoS data for experimental services

Category	Web service	Avg. performance	Avg. availability
Multiply	Multiply Slow	0.493	1
	Multiply Available	0.243	0.64
	Multiply Standard	0.239	1
Power	Power Slow	0.539	1
	Power Available	0.231	0.65
	Power Standard	0.225	1
Add	Add Slow	0.543	1
	Add Available	0.253	0.67
	Add Standard	0.246	1

In this case, there are 27 possible compositions that could be built. However, each composition will have different value for the entire quality and the platform should select the one that has the highest score. Figure 4 presents a graphics of the calculated values for the quality of compositions for the possible combinations. To run this simulation we have set the weights for performance and availability to 0.5 and 0.5. The composition on the top of the graphics has highest score compared to the rest. This is the composition *{Multiply Standard, Power Standard, Add Standard}*.

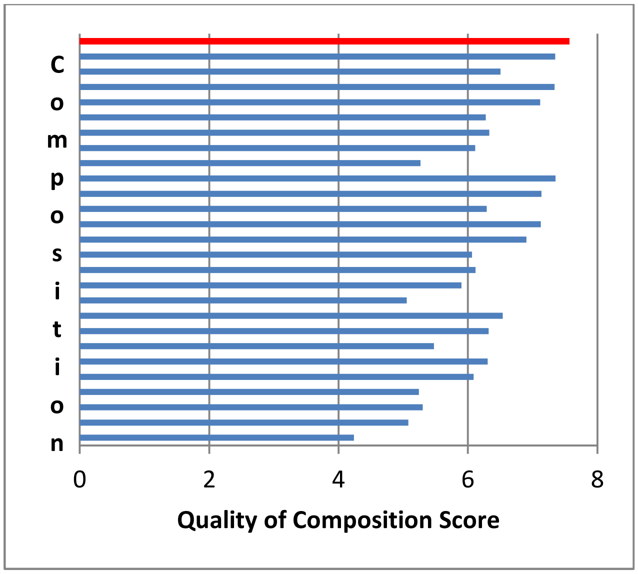


Figure 5. Scores for the Quality of Analyzed Compositions

When the experiment started our platform analyzed the defined business process and the associated quality goals. Based on the defined web service categories the relevant service implementations were discovered and the final composition was set. Table 2 presents the web services that were selected as a result of our experiment QoS data for selected services after experiment.

TABLE II. QOS DATA FOR SELECTED SERVICES AFTER EXPERIMENT

Category	Web service	Avg. performance	Avg. availability
Multiply	Multiply Standard	0.239	1
Power	Power Standard	0.225	1
Add	Add Standard	0.246	1
Total Quality		0.71	1

The total time for the execution of the business process is 0.71 seconds and the availability remains 100%. This is the best possible composition that fits the predefined quality requirements and the associated weights for each of them.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a platform for autonomous web service compositions. This platform is able to monitor the web services execution and gather data for evaluation of service quality characteristics. This data is available through web service registry extension. The paper also proposes a model for analysis of the quality data and determining the best service composition. A key aspect of this model is the introduction of *web service category* as an abstract way of defining a set of services providing the same functionality and interface.

In addition, our platform is based on open source software and is designed for easy extendibility and modifiability. In the long term such an approach could save a lot of administrative work and increase the level of customer satisfaction. The platform provides means for autonomously adapting the running business processes based on predefined user goals in terms of SLA. However, we can state the following directions for future research, in order for the platform to provide a fully functional end-to-end solution:

1. **Extending the scope of the web service interceptor** – currently, we support Apache Axis2 based web services but we plan to develop interceptors for other web service frameworks that can be extended.
2. **Extending the number of quality attributes** – currently, we have focused our research on *performance*, *availability* and *throughput*. We consider extending the number of supported quality attributes within the interceptors and the extended web service registry.
3. **Improving the model for selecting best web service composition** – a weak point for our model is the selection of the best web service composition. It is expected that all possible compositions are analyzed and then the best one is selected. As a point of improvement, we consider optimizing the selection algorithm to work in a more efficient way.
4. **Perform detailed validation of the platform –** The presented platform is still a prototype. As the platform gets more mature, additional validation and experiments should be performed.

ACKNOWLEDGEMENT

The research presented in this paper was partially supported by the DFNI I02-2/2014 (ДФНИ И02-2/2014) project, funded by the National Science Fund, Ministry of Education and Science in Bulgaria.

REFERENCES

[1] A. Silva, H. Ma, and M. Zhang, “A GP Approach to QoS-Aware Web Service Composition and Selection”, G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, 2014, pp. 180–191.

[2] A. Machado and C. Ferraz, “Guidelines for performance evaluation of web services”, In Proceedings of the 11th Brazilian Symposium on Multimedia and the web (WebMedia '05), Renata Pontin M. Fortes (Ed.). ACM, New York, NY, USA, 2005, pp. 1-10.

[3] L. Bass, P. Clemens, and R. Kazman, Software Architecture in Practice, Addison Wesley, 2013.

[4] K. Baylov, D. Petrova-Antonova, and A. Dimov, “Web service QOS specification in BPEL descriptions”, In Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14), Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, 2014, pp. 264-271.

[5] H. Birgit and S. Marchand-Maillet, “Rank Aggregation for QoS-Aware Web Service Selection and Composition”, 6th IEEE International Conference on Service-Oriented Computing and Applications, 2013, pp. 252-259

[6] B. Buhnova, S. Chren, and L. Fabriková, “Failure data collection for reliability prediction models: a survey”, In Proceedings of the 10th international ACM Sigsoft conference on Quality of software architectures (QoSA '14). ACM, New York, NY, USA, 2014, pp. 83-92.

[7] R. Douglas, D. F. Pigatto, J. C. Estrella, and K. Branco, “Performance evaluation of security techniques in web services”, In Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services (iiWAS '11), ACM, New York, NY, USA, 2011, pp. 270-277.

[8] E. Blanco et al., “A Transactional-QoS Driven Approach for Web Service Composition”, Z. Lacroix and M.E. Vidal (Eds.): RED 2010, LNCS 6799, 2012, pp. 23–42.

[9] L. O'Brien, P. Merson, and L. Bass, “Quality Attributes for Service-Oriented Architectures”, In Proceedings of the International Workshop on Systems Development in SOA Environments (SDSOA '07), IEEE Computer Society, Washington, DC, USA, 2007, pp. 3-

[10] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “QoS-Aware Middleware for Web Services Composition”, IEEE Trans. Softw. Eng. 30, 5 (May 2004), 2004, pp. 311-327.

[11] L. Hideo, V. Nakamura, A. L. V. Cunha, J. C. Estrella, M. J. Santana, and R. H. C. Santana, “A comparative analysis of algorithms for dynamic web services composition with quality of service”, In Proceedings of the 19th Brazilian symposium on Multimedia and the web (WebMedia '13), ACM, New York, NY, USA, 2013, pp. 217-224.

[12] M. A. Serhani, R. Dssouli, A. Hafid, and H. Sahraoui, “A QoS Broker Based Architecture for Efficient Web Services Selection”, In Proceedings of the IEEE International Conference on Web Services (ICWS '05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 113-120.

[13] M. Fluegge, I. J. G. Santos, N. P. Tizzo, and E. R. M. Madeira, “Challenges and techniques on the road to dynamically compose web services”, In Proceedings of the 6th international conference on Web engineering (ICWE '06), ACM, New York, NY, USA, 2006, pp. 40-47.

[14] N. Aziz, J. Byun, and Y. Park, “A QoS-Aware Performance Prediction for Self-Healing Web Service Composition”, Second International Conference on Cloud and Green Computing, 2012, pp. 799-803.

[15] D. Petrova-Antonova and S. Ilieva, “DYSCO: A Platform for Dynamic QoS-Aware Web Service Composition”, IADIS International Conference on Theory and Practice in Modern Computing 2012, Lisbon, Portugal, July 17-19, 2012, pp. 91-94.

[16] D. Petrova-Antonova and A. Dimov, “A QoS Driven Approach for Probability Evaluation of Web Service Compositions”, 6th International Conference on Software and Data Technologies, Volume 1, Seville, Spain, 18-21 July, 2011, pp. 321-326.

[17] D. Petrova-Antonova, “Cost Dependent QoS-based Discovery of Web Services”, Proceedings of International Conference on Software, Services & Semantic Technologies, September 11-12, 2010, Varna, Bulgaria, ISBN 978-954-9526-71-4, 2010, p. 152-159.

[18] S. Ran, “A model for web services discovery with QoS”, SIGecom Exch. 4, 1 (March 2003), 2003, pp. 1-10.

[19] Y. Lu, Z. Gao, and K. Chen, “A Dynamic Composition Algorithm of Semantic Web Service Based on QoS”, Second International Conference on Future Networks, 2010, pp. 354-356.

[20] Y. Yu, H. Ma, and M. Zhang, “A Hybrid GP-Tabu Approach to QoS-Aware Data Intensive Web Service Composition”, G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, 2014, pp. 106–118.

[21] Y. Xia and Y. Yang, “Web Service Composition Integrating QoS Optimization and Redundancy Removal”, 20th IEEE International Conference on Web Services, 2013, pp. 203-210.

[22] Y. Liu, A. H. Ngu, and L. Z. Zeng, “QoS computation and policing in dynamic web service selection”, In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (WWW Alt. '04), ACM, New York, NY, USA, 2004, pp. 66-73.

[23] Z. Brahmi and M.M. Gammoudi, “QoS-Aware Automatic Web Service Composition based on cooperative agents”, Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2013, pp. 27-32.

[24] Z. Zheng and M. R. Lyu, 2013, “Personalized Reliability Prediction of Web Services”, ACM Trans. Softw. Eng. Methodol. 22, 2, Article 12 (March 2013), 2013, pp. 1-25

[25] J. Bih, 2006, “Service oriented architecture (SOA) a new paradigm to implement dynamic e-business solutions”, *Ubiquity* 2006, August, Article 4 (August 2006), 2006, pp. 1-1.

[26] V. Muthusamy, H. Jacobsen, T. Chau, A. Chan, and P. Coulthard, “SLA-driven business process management in SOA”, In *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research* (CASCON '09), Patrick Martin, Anatol W. Kark, and Darlene Stewart (Eds.). IBM Corp., Riverton, NJ, USA, 2009, pp. 86-100.

Balancing Centralised Control with Vehicle Autonomy in AGV Systems for Industrial Acceptance

Markus Bader

Vienna University of Technology
Institute of Computer Aided Automation (ACIN)
Vienna 1040, Austria
Email: markus.bader@tuwien.ac.at

Andreas Richtsfeld, Wolfgang Holl

DS AUTOMOTION GmbH
Technology & Product Development
Linz 4030, Austria
Email: [a.richtsfeld, w.holl]@ds-automotion.com

Markus Suchi, George Todoran, Markus Vincze

Vienna University of Technology
Automation and Control Institute (ACIN)
Vienna 1040, Austria
Email: [markus.suchi, george.todoran, markus.vincze]@tuwien.ac.at

***Abstract*—Automated Guided Vehicle (AGV) systems have to fulfil safety requirements and work reliably in order to be cost-effective and gain industry acceptance. Consumers want flexible AGV systems which require autonomous and distributed components to work, but this autonomy is often perceived as a disadvantage and a safety hazard. This work presents ongoing attempts and challenges to the distribution of knowledge and autonomy within AGV fleets while still ensuring safety and efficiency. Acceptance is gained by the integration of expert knowledge and a smoothly adjustable level of AGV autonomy which allows for a balance between centralized control and vehicle autonomy. Results are shown using a 3D physics simulation of a small production site.**

Keywords–AGV; Robotic; Industry; Safety; Planning; Autonomous; Navigation.

I. Introduction

Automated guided vehicles (AGV) are driverless mobile platforms primarily used for transportation processes as well as for flexible system solutions on assembly lines. Applications for AGV systems span from automated harbours where containers are moved around to pallet transport in warehouses. Hospitals use AGVs to automate processes, such as laundry and preparation of medication and to transport food and other goods between stations. The workspaces of humans and AGVs are normally separate but accessible to one another.

Prevention of collisions and deadlocks is imperative, and regular tasks, such as recharging or vehicle cleaning, must be managed. Reliability and safety are important issues, therefore simple and straight forward approaches are preferable. Thus, AGV systems are mainly designed off-line, with manually designed tracks, sometimes only for one specific vehicle type to make on-board planning obsolete. This is done to simplify centralised coordination and to enable an efficient overall control process.

Most AGV systems are individually designed for a certain application, which generates a market for highly specialized companies. Kiva Systems [1], with its AGVs for warehouse automation, is one of the most well-known companies. DS-Automotion [2], the project's partner, produces AGV systems with similar technology, but in contrast to Kiva Systems, upgrades a variety of vehicle types, ranging from small self-made platforms for the automotive or health-care industries to standard transportation products, such as forklifts. In addition, logistic solutions are provided for health-care, paper, and automotive industries, as well as for intra-logistic applications. Figure 1 shows some of these AGVs.

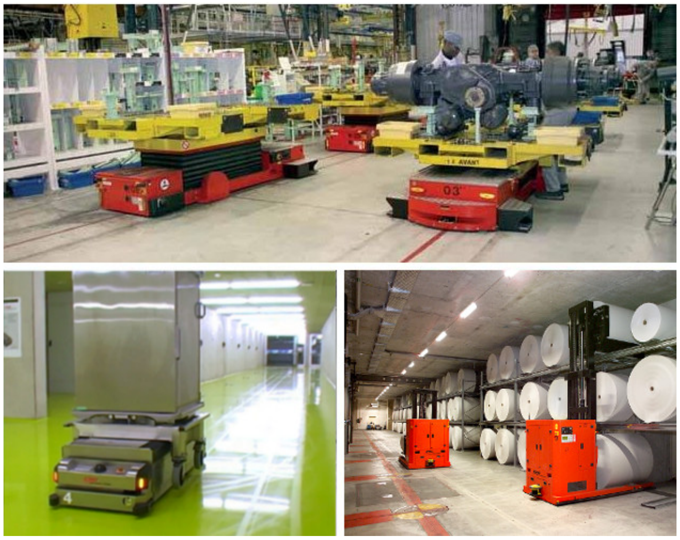


Figure 1. DS-Automotion's AGVs in action on an automotive assembly line, in a hospital and a paper factory.

In the last few years, customers have been increasingly requesting flexible and customisable solutions. They want systems to operate in environments with humans and they do not want to reconstruct their (often leased) buildings in order to accommodate an AGV system.

Normally, AGVs are not autonomous agents. This means

ISBN: 978-1-61208-405-3

that all vehicles of a fleet are guided by a centralised system which supervises the overall transport process. The agent's autonomy is limited to safety actions to ensure a safe overall process. A more flexible solution would lead to more data to process and higher computational costs. This additional data would not be manageable in real-time by a centralised system, as the bandwidth and the computational costs would be too high. As a result, control must be distributed and agents have to gain more autonomy in making decisions. However, autonomous agents are not well accepted in industrial applications and therefore a balance has to be struck among demands, flexibility and control. This paper targets exactly this problem by proposing a hybrid system which is able to scale the level of autonomy for each vehicle on demand and integrate expert knowledge into the system.

Section II describes a typical AGV system and the state of the art in mobile robotics. Our approach is presented in Section III and the challenges to face in Section IV. Results are shown in Section V, followed by a conclusion.

II. STATE OF THE ART

The structure of a classical AGV system is depicted in Figure 2 and works in the following way: The *AGV control system* (ACS) is driven by requests from the *Production Planning and Control* (PPC) module which disassembles general processes into internal processes. General processes are externally triggered processes such as customer requests, in contrast to internal processes, which describe the processes needed to fulfil externally triggered processes. Operation orders for AGVs are therefore part of internal processes and must be coordinated. The ACS assigns operation orders to the vehicles, specifies the track the vehicles have to follow and controls their speed in order to avoid collisions and deadlocks.

A. Industry

The automation industry prefers straightforward and non-complex solutions. For example, magnetic or RFID markers under the real, physical predefined track are commonly used for localization as well as for path-planning. The agent's on-board tracking control has to simply follow the *bread crumbs* of marker beacons. Such a control typically takes advantage of a flat system output [3], which in this case is the robot pose performing the tracking control. The benefit of such *bread crumb* localization is the low computational costs needed for localization and for trajectory planning. This type of navigation is sufficient for many industrial applications. Expensive safety-certified sensors and controllers are required, if there are humans in the same workspace. In this case, safety controllers must be used to override the motor controller commands in order to prevent accidents. SICK [4] produces certified laser range scanners which are able to dynamically adapt the safety areas to the vehicle's velocity. Obstacles detected within a safety area cause an emergency halt. An emergency halt means that an agent has to move itself into a safe state and cannot just stop moving, e.g.,, the system has to prevent agents from stopping in front of an emergency exit. Laser range sensors are thus mounted on AGVs in order to detect obstacles. However, the lasers are not necessarily used for navigation because of the additional complexity required. This forces every AGV to stay on the predefined tracks, therefore leaving a track in the case of an obstacle is not possible. An obstacle on the track will cause the AGV to slow down and eventually to stop. Even if localization techniques are able to deal with deviation from predefined tracks, they are usually avoided in order to keep systems simple.

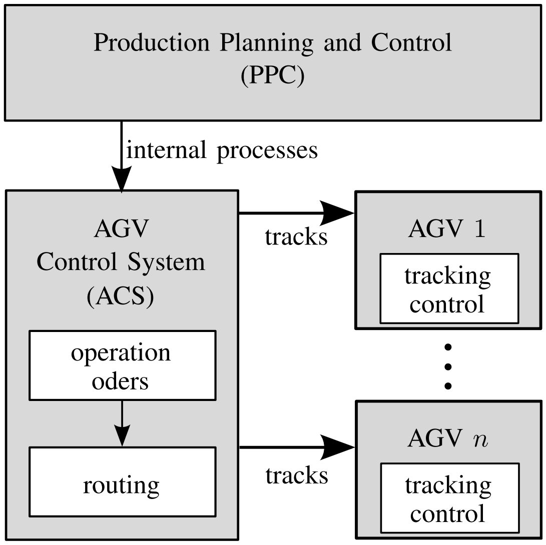


Figure 2. Modules of a classical AGV-system. A single server routes all AGVs along offline defined tracks; no path planning is involved.

DS-Automotion controls its AGV fleet by dividing the tracks into segments of arcs and lines. The control system distributes to each robot the next course segments to follow. This enables the ACS to prevent collisions by exclusively assigning only one segment at a time per agent. Due to the complexity of this problem, the time frame is limited and heuristics need to be used in order to reduce computational complexity. The goal of the approach proposed here is to decouple routing and local planning. The AGV should be able to recognize specific scenarios and should deal with them locally by adapting its trajectory planning strategy accordingly.

B. Research

The IEEE Robotics & Automation Magazine, Special Issues March 2014, recently summarised the state of the art and research done on perception and navigation for autonomous vehicles with articles on dynamic environments [5], risk analysis [6], self-localization and mapping in in- and outdoor environments [7], object recognition as well as path-planning [8] and motion-planning [9]. All of these research topics have to be combined to create an autonomous vehicle fleet. Projects, such as the DARPA [10] challenges have successfully demonstrated this, but the commercial market still lacks reliable autonomous agents.

Since 2014, Robot Operating System (ROS) has offered a software package dealing with AGVs [11]. The code collection includes drivers and simulations for an Ackermann type robot intended for logistics transport. The framework enables users to define tracks using waypoints, and the simulated AGV is able to follow these tracks. The ROS navigation stacks [12] are used to control and localise the vehicle. A logistic framework to coordinate multiple AGVs is missing, and the system is not able to deal with expert knowledge. However, we believe that this expert knowledge is vital for the commercial market and for industrial acceptance.

Similar set-ups to AGV systems can be found in RoboCup [13]. Competitions like RoboCup's Small Size League (SSL) soccer is designed to improve multi-agent cooperation through friendly competition.

The environment has similarities to an AGV system. Agents are controlled by a centralised server, and all objects on the playing field are tracked by a standardised vision system. The league has shown that it is possible to detect specific scenarios and to react quickly by adapting *plays* [14]. A play denotes a sequence of actions or behaviours according to a playbook, e.g., follow track and slow down. The playbook describes recognizable scenarios with according plays, e.g., a scenario (with an automated fork lift) in front of an elevator door ⇒ play: verify that the fork is folded before entering the lift; follow track precisely with low speed. Plays can also provide predefined plans with roles for multiple agents which can be adapted to scenarios to prevent deadlocks or collisions, for example in the following scenario: the passing of two agents ⇒ play: select leader; leader selects side for passing; follower acknowledges side; passing. Similar techniques are also used in RoboCup's Middle Size League (MSL), which has no centralised command system.

In the approach proposed here we are presenting an idea for how to integrate expert knowledge into the system to support play selection. This is done by augmenting track segments as well as areas around segments in order to simplify scenario recognition and to enable reproducible behaviour.

III. Approach

AGV systems currently deployed in industrial applications use manually offline designed tracks for path planning. These tracks are defined by a list of segments and distributed by the ACS to the AGVs, as shown in Figure 2. An AGVs task is to follow these segments. This system has only two planning levels:

- the overall routing on the centralised server and
- the on-board tracking control on the AGV.

Obstacles on the track always trigger an emergency halt.
We would like to present an approach which enables an AGV system to additionally:

- autonomously avoid obstacles on the track,
- solve situations without the ACS interfering, e.g., a multi-robot situation or pick and place actions,
- use optimised trajectories to be time-, energy- and/or resource-optimal (e.g., floor abrasion), and
- be easier to maintain and less expensive during system design and set-up.

This can only be realized if AGVs are able to:

- localise themselves, (even when leaving the predefined track),
- communicate with each other, and
- execute and adapt their behaviour (role play), to solve local issues without centralised intervention.

We propose that the ACS distribute segments to the AGVs, similar to before, but encapsulated with additional attributes. For demonstration purposes we will group areas into *free* or *critical*. A free area signals that an AGV is allowed to leave the track; a critical area indicates particular precaution. The additional segment attributes are used to indicate to the system what to expect or which behaviour (role play) should be selected to manage the track segment. A typical attribute would be that no stops are allowed. This is important when passing a fire door: the agent would have to verify whether there is enough space free after the critical section before entering said section and would select an appropriate motion control algorithm. In our approach, agents are able to select one of two motion control algorithm.

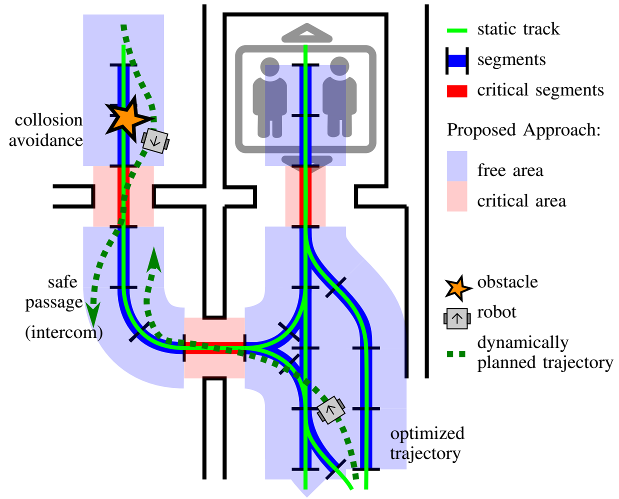


Figure 3. This figure depicts the limitations of the a classical AGV system and the advantages to be gained by the new approach proposed.

- A *Model Predictive Control* (MPC), implemented similarly but in a more advanced way than the *Dynamic Window Approach* (DWA) [15] to follow tracks, which allows the system to diverge from the track and to avoid obstacles.
- A tracking controller based on a flat system output [3], which tries to follow tracks precisely. In the presence of an obstacle, the control slows the vehicle down, eventually stopping it.

Figure 3 shows the limitations of the old approach and the advantages of the new approach: The system currently used has centralised path planning based on predefined line and arc segments (blue). An AGV has to follow the static tracks (green) routed by the control system. In the face of an obstacle, the AGV slows down and eventually stops on the track.
In contrast to the system currently used, the system proposed here uses predefined areas in which a vehicle is allowed to move freely. Obstacles can be circumnavigated and two or more vehicles are able to directly communicate in order to plan trajectories for safely passing each other. Trajectories are locally planned and can be time-, resource- or energy-optimised.

A. First Iteration – Behaviour Controller, Role Play and Playbook

The aforementioned concepts *play/role play* and *playbook* describe strategies for facing specific scenarios. A play or role play describes the parameter selection, and an interaction procedure for one or more robots for a certain length of time. The playbook holds multiple such predefined role plays for selection. The behaviour controller implemented is in charge of recognising scenarios and selecting appropriate plays. In the first iteration, simple role plays are implemented with the goal of getting a system up and running as it was before. AGVs use the predefined tracks as a basis for local path planning, but they may change their local path when an obstacle is blocking it or when indicated to by predefined segments. Agents are able to select between two tracking control types for different motion behaviours: MPC or low level tracking control. The controller parameters are selected on-demand depending on the role play executed. This enables the vehicle to behave differently in different areas, while also giving the operator the capability of restricting the system, when necessary.

B. Second Iteration – Robot-Robot interaction

The second stage will enable vehicles to plan their own paths, if permitted within the current area. Using the aforementioned playbooks with role plays for specific scenarios enables the control system to detect such scenarios and to initiate role plays with one or more agents involved, e.g., the passing of two vehicles in a hallway or passing a door. If such a scenario is recognised, the vehicles involved are allowed to communicate with each other to adapt the known role play. This allocates the control competences to the agents, thus making the system more flexible.

C. System Components

The overall architecture of the system proposed is shown in Figure 2 and in more detail in Figure 4. A PPC module coordinates the overall process and interfaces the company's accounting system, e.g., an *Enterprise Resource Planning* (ERP) system. The ACS gives transportation orders to the vehicles *(job planner)* and plans optimal routes for each vehicle *(route planner)*. Each AGV implements a *Behaviour Controller* (BC) as a state machine, which makes binary decisions for them and selects role plays. The BC module controls the AGV while autonomously solving situations based on the aforementioned playbook and communicates success or failure to the ACS. It triggers local navigation modules if a new plan needs to be computed.

IV. CHALLENGES

In our approach, we introduce two fundamental changes to the AGV system currently used by DS-Automotion. Each vehicle has a navigation module with a path-planner and a motion controller (*aka* a trajectory-generator), as well as a behaviour controller (BC) to trigger role plays.

A. Navigation

Two navigation layers in the AGV are used in the system proposed, namely *path planning* and *motion control*, which are often denoted as global and local planning, respectively. This may cause confusion because the centralised ACS also has a planning module which computes the overall *AGV routing tables* for the fleet.

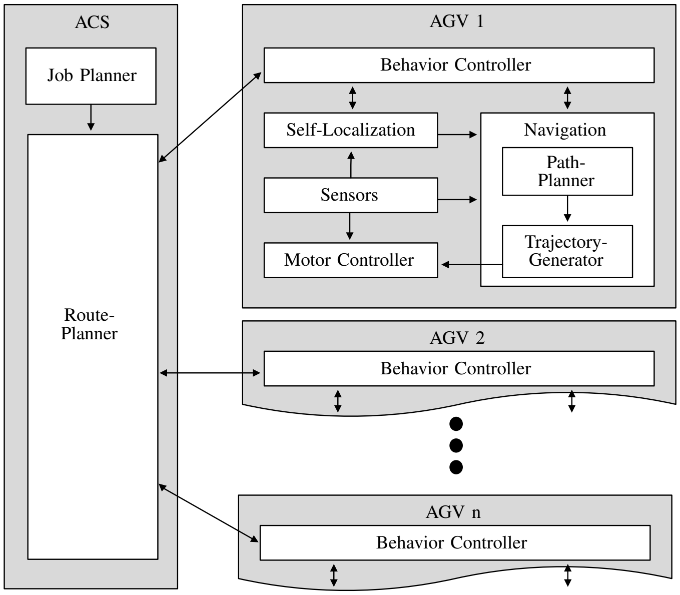


Figure 4. AGV system overview: The ACS gets orders from the PPC (see Figure 2) and distributes them to the AGVs. The ACS also supervises AGV route planning in order to optimise the execution time of all of the orders given to the system.

1) AGV routing tables: In order to take full advantage of the vehicles' capabilities, the ACS has to be adapted. The route planning has to cope with variances in execution time, as the execution time of a role play can vary due to local navigation. The ACS has to learn and adapt parameters such as execution time and success rates of role plays in order to create optimal routing tables.

2) Path Planning: The local path-planning receives routes to follow from the ACS and delivers segments to follow to the motion control. A challenge at this layer arises if a vehicle leaves the predefined tracks. The path-planner has to compute a suitable path by using known maps of the environment or to communicate an issue to the ACS. It is also the task of this planner to find paths to objects for pickup.

3) Motion Control: Virtually exact tracking control can be achieved by using, a flat output system model and track segments as splines as input , but the system proposed should be able to diverge from the track if needed. An MPC [9] generates a possible trajectory based on the current system state and weights each one based on a cost function which can include the obstacles detected. This control executes the first control sequence of the winning trajectory only until the next control iteration. Continuous updates are needed for safe and smooth motion control. It is commonly known that the most computationally intensive procedures in this cost function are collision detection and the evaluation of motion costs, but the latest research has demonstrated that the introduction of proper heuristics effects a huge performance gain [16], [17]. In our approach, we allow the system to switch between an exact tracking control and the MPC.

B. Behaviour Controller

Playbooks as used in RoboCup soccer scenarios must be developed to simplify plans, especially when multiple agents are involved. The selection of a lead agent during a multi-agent role play must be managed [18]. However, more important for the acceptance of the system are the integration of expert knowledge and reproducible behaviours.

C. Self-Localization

Another challenge occurs due to the inaccuracy of self-localization when using laser-based localization methods, such as *Adaptive Monte Carlo Localization* (AMCL) [19], which is implemented in the ROS or in the *Mobile Robot Programming Toolkit* (MRPT) [20]. The system has to deal with inaccuracies, and eventually has to adapt its behaviour to gain a better localization confidence when needed. For example, a pick and place procedure where one vehicle places a payload and another vehicle picks it up fails upon inaccurate localization.

D. Mapping

In order to be cost-effective, AGV systems with customised vehicles are usually deployed for a long period of time. During this long period of use, changes to the environment can be expected and must be dealt with. A common map layer which represents daily changes to the environment can be updated and distributed to the vehicles. Creating a sound map of multiple measurements is a difficult challenge and it is not yet clear whether this task should be performed by each agent individually or by a centralized unit, especially if loop closing is necessary.

E. Industrial Acceptance

Industry demands flexible and easily maintainable solutions. This is only manageable with a distributed system, but that inherently increases the system's complexity. It will be a challenge to find the right balance and in this study, suitable role plays with expert knowledge in order to create an acceptable system for industry.

V. Results

We are currently at the first iteration level, as proposed in Section III with a simple working set-up.

A. Set-Up

We interfaced the ACS used by DS-Automotion to intercept operation orders and computed routing tables. The ACS has multiple safety features to ensure a safe process. For example, all vehicles are monitored to verify that vehicles are on the track following the assigned route. A simulated environment was created using GazeboSim [22], a freely available 3D simulation package including a physics engine. We simulated a production site within our lab with multiple vehicles. Figure 5 shows two related snapshots. The MRPT-library is used for localization and ShmFw [23] for communication and visualization of data. ShmFw is a fast dynamic framework based on the boost inter-process library [24], which uses shared memory elements for inter-process communication. ROS libraries are only used to interface the simulator by using customized ROS nodes to exchange data between ROS messages and ShmFw variables. The decision to avoid ROS in the functional code is due to the current system used by the project partner, who uses their own middleware. Another reason to exclude ROS was down and upward compatibility. Compared to the product cycles of AGVs, the release cycles of ROS are very short, and usually a company has to support products for many years. However, there is an industrial version called ROS-Industrial [25], which might be of interest for future projects.

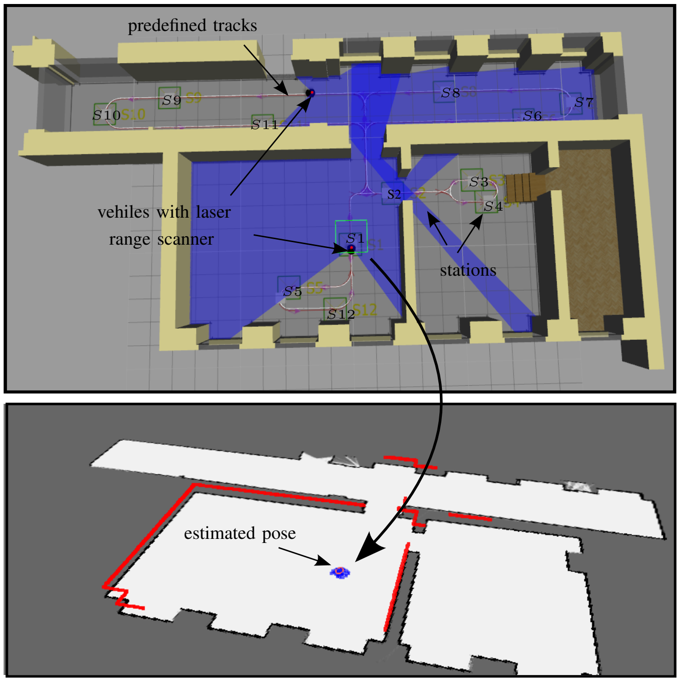


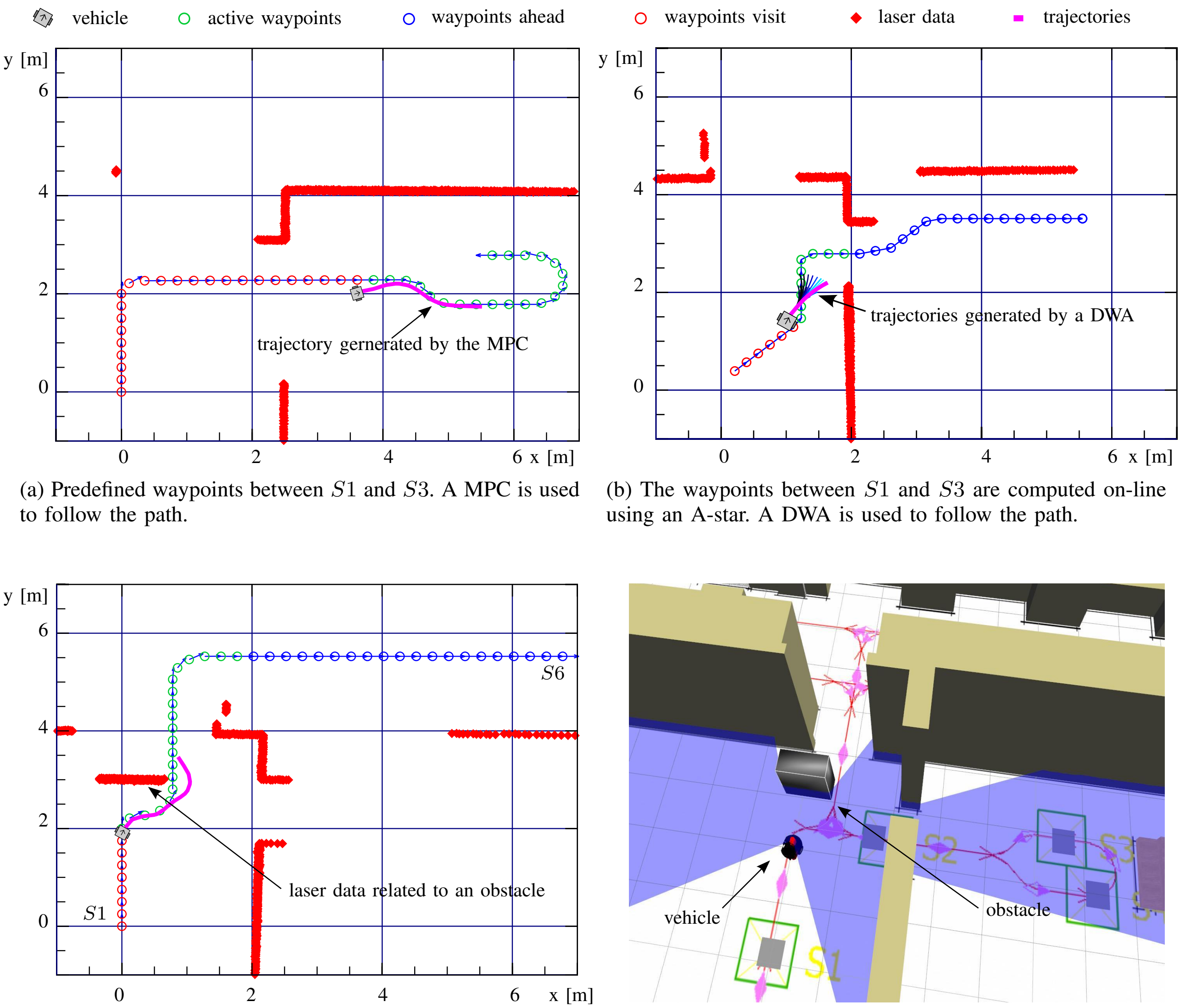
Figure 5. Top: Simulated production site with tracks, stations $S1 - S12$ and two simulated vehicles with a SICK laser range scanner and rays in blue. Bottom: MRPT particle filter self-localization with an estimated robot pose on a previously-generated map using a Rao-Blackwellised particle filter SLAM, also implemented in MPRT.

B. Autonomy

At the current level we are able to start vehicles at arbitrary locations. The system uses a local path planner to find a path to the next known track to receive orders. The aforementioned complex initialisation procedure is still needed because of safety issues. All segments delivered to vehicles are augmented with additional parameters to trigger different behaviours, such as switching between the multiple implemented motion control methods with various settings. The operator is now able to predefine areas to control vehicle behaviour in advance. For example, vehicles in open areas use an MPC or DWA to cope with blocked tracks. In areas such as turns between $S3 - S4$, shown in Figure 5, which are close to the stairways, the operator is able to predefine a motion control to follow the track as precisely as possible. Figure 6 shows cases with unblocked and blocked paths, as well as different tracking controls implemented. The behaviour shown in Figure 6c allows the vehicle to select trajectories to avoid collisions with obstacles next to the path, but an obstacle on the path would cause the vehicle stop. This behaviour was designed to increase the acceptance of the system and was favoured by the industrial project partner.

C. Path-planning

The waypoints shown in Figure 6a and 6c are based on the static predefined segments and placed at a constant distance to represent the path, in contrast to Figure 6b. In this case the AGV computed its own track by taking the previously



(a) Predefined waypoints between $S1$ and $S3$. A MPC is used to follow the path.

(b) The waypoints between $S1$ and $S3$ are computed on-line using an A-star. A DWA is used to follow the path.

(c) The MPC's cost function includes sensor readings and avoids collisions with an obstacle next to the path.

(d) Simulated environment related to Figure 6c. The obstacle next to the path would cause a collision.

Figure 6. Operation orders executed with different behaviours.

recorded map and the start and goal position into account. The trigger for doing so is based on the expert knowledge encoded in the track segments delivered by the ACS. The route is not as smooth as the one delivered by the ACS, but in this case the DWA implemented takes care of this problem and avoids obstacles by using a cost function to weight possible trajectories within a certain time window, as shown in Figure 6b. In this way the ACS does not have to take care of replanning until the AGV signals otherwise. The current system is now able to perform at the same level of efficiency as before but is also able to cope with obstacles.

VI. Conclusion

This work presents a recently-begun research project with the goal of transferring research knowledge from the field of mobile robotics to the industrial application of AGV systems. We proposed an approach for decentralisation of the control system in order to achieve a flexible solution. The approach entails enhancing agents with an on-board self-localization and navigation module as well as a behaviour controller for carrying out autonomous actions. The centralised control system has to deal now with autonomous agents, shifting the task from control of them to coordination of them. Expert knowledge augments, on the one hand, the map to allow or confine autonomous actions in specific areas, and on the other hand, the route delivered to the AGV to prepare the agent for scenarios. We believe that only a balanced system tuned by the human operator on site will be accepted in industrial applications, and reproducible behaviour, as well as human ability to influence autonomous behaviour are vital to this acceptance.

Acknowledgment

The research leading to these results has received funding from the Austrian Research Promotion Agency (FFG) according to grant agreement No. 846094 (Integrierte Gesamtplanung

mit verteilen Kompetenzen für fahrerlose Transportfahrzeuge).

References

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses," in Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2, ser. IAAI'07. AAAI Press, 2007, pp. 1752–1759.

[2] "DS-Automotion," 2015, URL: http://www.ds-automotion.com [accessed: 2015-04-04].

[3] M. Liess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," International Journal of Control, vol. 61, no. 6, 1995, pp. 1327–1361.

[4] Detection and Ranging Solutions, SICK AG, Erwin-Sick-Str. 1, 79183 Waldkirch, Germany, December 2011, 8014402/2011-12-20. [Online]. Available: http://www.sick.com

[5] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized centroid and formation control for multi-robot systems," in Robotics and Automation (ICRA), 2013 IEEE International Conference on, May 2013, pp. 3511–3516.

[6] C. Laugier et al., "Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety," Intelligent Transportation Systems Magazine, IEEE, vol. 3, no. 4, Winter 2011, pp. 4–19.

[7] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, vol. 1, Sept 2003, pp. 842–849.

[8] S. M. LaValle, Planning Algorithms. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[9] T. Howard, M. Pivtoraiko, R. Knepper, and A. Kelly, "Model-Predictive Motion Planning: Several Key Developments for Autonomous Mobile Robots," Robotics Automation Magazine, IEEE, vol. 21, no. 1, March 2014, pp. 64–73.

[10] S. Thrun et al., "Stanley: The robot that won the DARPA Grand Challenge," Journal of Field Robotics, vol. 23, no. 9, 2006, pp. 661–692.

[11] "ROS: The agvs package," 2015, URL: http://wiki.ros.org/agvs [accessed: 2015-04-04].

[12] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The Office Marathon: Robust navigation in an indoor office environment," in Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 300–307.

[13] H. Kitano, M. Asada, I. Noda, and H. Matsubara, "RoboCup: robot world cup," Robotics Automation Magazine, IEEE, vol. 5, no. 3, Sep 1998, pp. 30–36.

[14] J. Bruce, M. Bowling, B. Browning, and M. Veloso, "Multi-robot team response to a multi-robot opponent team," in Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, vol. 2, Sept 2003, pp. 2281–2286.

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," Robotics Automation Magazine, IEEE, vol. 4, no. 1, Mar 1997, pp. 23–33.

[16] R. Knepper and A. Kelly, "High Performance State Lattice Planning Using Heuristic Look-Up Tables," in Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, Oct 2006, pp. 3375–3380.

[17] M. Suchi, M. Bader, and M. Vincze, "Meta-Heuristic search strategies for Local Path-Planning to find collision free trajectories," in Proceedings of the Austrian Robotics Workshop (ARW-14), May 2014, pp. 36–41.

[18] N. Basilico, N. Gatti, and F. Amigoni, "Leader-follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies," ser. AAMAS '09. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 57–64.

[19] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, 2005.

[20] "Mobile Robot Programming Toolkit (MRPT)," 2015, URL: http://www.mrpt.org [accessed: 2015-04-04].

[21] "RISC Software GmbH," 2015, URL: http://www.risc-software.at [accessed: 2015-04-04].

[22] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 3, Sept 2004, pp. 2149–2154.

[23] "ShmFw," 2015, URL: https://github.com/ShmFw/shmfw [accessed: 2015-04-04].

[24] "Boost.Interprocess," 2015, URL: http://www.boost.org/doc/libs/1_55_0/doc/html/interprocess.html [accessed: 2015-04-04].

[25] "ROS-Industrial," 2015, URL: http://http://rosindustrial.org [accessed: 2015-04-04].

Model-Driven Architecture for Self-Adaptive Context-Aware Message Routing in Pervasive Environments

Nachoua Guizani
SFR Santé Lyon-Est, ERIC Lab
Université Lyon 1, Université Lyon 2
Lyon-Bron, France
e-mail: nachoua.guizani@gmail.com

Jocelyne Fayn
SFR Santé Lyon-Est : eTechSanté
INSERM US7, Université Lyon1
Lyon, France
e-mail: jocelyne.fayn@inserm.fr

***Abstract*—Ensuring suitable message transmission Quality of Service (QoS) still remains among one of the most crucial requirements, especially in case of emergency. A message routing policy application should face the challenge of dynamic source and target context changes, such as resources availability and environmental conditions, and adapt its behavior and decisions accordingly. In this paper, we propose a context-aware model and a functional architecture aimed to steer an intelligent, personalized and adaptive message routing policy. Our objective is to enhance at the operational level message transmission QoS in terms of delivering the right message to the right recipient with the right delay requirements, by taking into account message, sender and recipient ecosystems. In the proposed architecture, we highlight how a message routing policy can reason about context information and adapt autonomously its behavior in response to unpredictable events and context changes in pervasive environments. This architecture is based on ambient intelligence and complies with different scenarios. The relevance of our approach is demonstrated by a use-case in the eHealth domain.**

Keywords-context aware systems; adaptive system; ecosystem; message routing policy; Ambient Intelligence.

I. INTRODUCTION

Various applications in the computing field perform multiple tasks, which may require a reliable coordination and communication between different actors. These tasks produce a workflow chain embedding a large number of messages which have to be properly routed and timely handled, especially in emergency scenarios. Several emergency real-life situations are subject to quite high failure rates because of poor communication infrastructures and uncontrolled and non-adaptive message routing policies. Messages are usually blindly transmitted to remote recipients without prior knowledge of their contextual environment availability, reliability and capability. In addition, in some cases the relevance of the recipients should be checked with respect to the urgency and to the message context in order to efficiently and rightly forward it. Several context changes can occur at run-time caused by mobility of users and devices especially in pervasive environments. To overcome these difficulties, we need to set up reliable and adaptable message routing policies ensuring an intelligent message exchange. This policy has to deal proactively with unpredictable events, such as recipient unavailability and exceeding the message treatment deadline, to continuously take into account context changes (e.g., climate, localization, etc.) and to adapt the message delivery accordingly.

The concepts of context-aware [1] and adaptive systems [2] are among the most exciting topics in ubiquitous computing [3] today. To achieve high levels of awareness and adaptivity in message exchanges among two different environments, the challenge is threefold: model source and target contextual information, identify the constraints called adaptation situations to which routing policy applications are sensitive [4], and adapt routing policy behaviors according to context changes.

In this paper, we propose, as a first step, a context-aware routing policy model showing the different objects involved in the message routing processes, as well as the relationships between them. Moreover, we propose a functional architecture of a distributed, context-aware system dealing with message routing policy management in pervasive environments. This architecture is based on AmI ambient intelligence [5] technologies.

Our main objective is to ensure, at operational level, message transmission Quality of Service (QoS). We define QoS as the ability of delivering a given message to the right recipient while satisfying delay and context constraints (availability, experience, trust, etc.). Indeed, we focus on making the routing policy: (*1) Personalized*: in the way that it routes the message to the most relevant destination according to context analysis and, for each message, determines the required delay for reception, reading and reply; *(2) Context-aware*: routing policy application is sensitive to context; (*3) Adaptive*: means that it is able to adapt its behavior in real-time according to context changes; *(4) Intelligent*: in the sense that we take advantage from AI technologies to integrate some intelligence in the routing policy decisions. To summarize, we aim to empower systems to autonomously deliver the right message to the right individual with the right delay requirements, by taking into consideration message, expeditor and recipient contexts.

The paper is further structured as follows. The next Section discusses related work. In Section 3, we present an UML model describing a context-aware message routing policy. In Section 4, we propose a context-aware architecture ensuring intelligent routing policy management. Section 5 shows a case study in the eHealth domain illustrating the message routing policy behaviors in an emergency scenario.

II. RELATED WORK

Several recent research papers spotlight workflow exceptions handling and routing policy management. [6] proposes a proactive detective control model to prevent possible shutdown and violations in workflow applications, and highlights the capacity of Service Level Agreement to ensure/provide QoS avoiding cloud services composition failure and improving the dynamicity of workflow execution. [7] takes advantage of context-aware systems to solve wireless local area network routing problems. However, in this paper the authors restrict the context to device energy-oriented context. The same yields for [8], which proposes an adaptive QoS and energy-aware routing approach for Wireless Sensor Networks (WSNs) based on an improved ant colony algorithm. Several other papers also demonstrate that efficient workflow management under unpredictable events effectively contributes to QoS improvement at different levels and in different fields. A typical example in the eHealth domain can be found in [9]which proposes a framework for modeling context-aware workflow driven resource allocation based on Petri Nets.

Several works have also addressed the context-aware computing paradigm, which becomes an important research issue especially with the emergence of ubiquitous computing [3]. According to [1], context-aware systems are a category of systems that adapt their behavior at run-time according to their users' needs, by proactively anticipating the users' needs without explicit user intervention. To deal with decoupling applications from context information layers, several middleware have been proposed in the computing literature. CAMidO [10] is a Context-Aware Middleware based on Ontology. The particularity of this middleware was to provide a metamodel for context description. The author's idea was to monitor significant context changes and consequently, to dynamically adapt the applications to react to these changes. Both, collection and adaptation are carried out based on an ontology representation. Context-Aware Middleware based on a context-awareness Meta-Model [4] is another middleware and run-time model for dynamic context management based on a model-driven architecture. The paper shows how applications can dynamically adapt their behavior at run-time according to context changes. [11] proposes Unified Context-Aware Application Model, a generalized context-aware architecture for heterogeneous smart environments. The context representation is ontology-based and deals with the 6 types of questions: Who, What, Where, When, Why and How.

Obviously, context-aware approaches have also been adopted in autonomic computing and self-adaptive systems design [2]. Self-adaptive systems aim at ensuring dynamic behavioral adaptation with respect to context changes. Adaptation in the computing literature can operate on four elements: service, interface, content, and software components. [12] proposes a dynamic adaptive service dealing with both highly dynamic changes in pervasive environment and limited resources. Three major steps shall be followed when designing self-adaptive systems: adaptation modeling, analysis and validation. [13] proposes a context Petri Net model for improving the correctness of the configuration of self-adaptive systems aimed at verifying reachability and liveness as key priorities. [14] combines Aspect-Oriented Models and run-time models to design an adaptation model for correct system configuration processing at run-time. The adaptation model includes a set of adaptation rules, which have been introduced to change the system behavior during execution.

Unfortunately, the issue of performing intelligent, adaptive and personalized routing policies has not been adequately treated by the presented research works. Furthermore, the multidimensional aspect of workflow management has not been taken into account. In the architecture we are proposing in this paper, we have taken these issues into consideration and we will build on AmI [5] to ensure a reliable message routing policy aware of the message, sender and receiver contexts.

III. CONTEXT-AWARE ROUTING POLICY MODEL

In this Section, we present our proposed context-aware message routing policy model (see Figure 1), as well as the terminology that we use in this paper. The model points out several concepts involved in message routing processes and the relationship between them.

As shown in Figure 1, a message routing process involves several entities: source actor also called message sender, target actor named receiver, and the message itself. Each entity is surrounded by its own environment and situations under which the message routing policy may change its behavior. A routing policy should provide different techniques and rules necessary to ensure data transportation from their source point to a target point. We distinguish two types of parameters required for message routing.

1) Preprocessing parameters: they should be known before message routing, such as the nature of the required destination, the message routing means (PC, phone, etc.) and type (SMS, mail, etc), etc.

2) Processing parameters: they are determined to control the routing process, such as the required delay for message reception, reading and reply.

The model reflects the multidimensional aspect of the message exchange problem. It highlights the three ecosystems to which the routing policy is sensitive: the source, target and message ecosystems. An *ecosystem* is a set of complex and scalable information systems that are related to entities in a given environment. The routing decision making should be driven by ecosystem data. As defined by [15], *an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".* In our case, we consider the entity as a physical or a logical element represented by a person, place or object which is involved in the message routing process. More generally, an *entity* in the message routing process can play several roles (sender, receiver, the concerned parties, author, etc.) in the triggering of an event (order, request, etc.), which generates a message transmission. Each message has its own ecosystem.

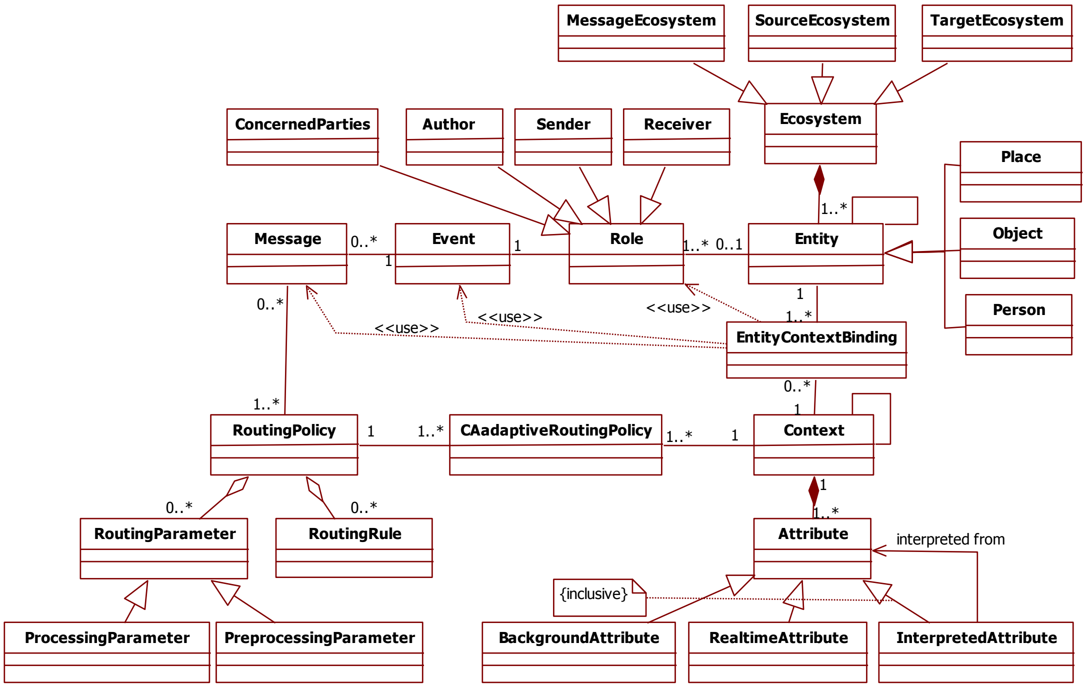


Figure 1. Context-aware message routing policy model

We distinguish different types of messages. Their content could be a piece of information or an alarm. They can be intentionally sent on request of a person, or automatically invoked or triggered by a device in case of a new event or of some failure. An ecosystem is composed of a set of entities.

An *entity* can be related to another. It can be related to a wide variety of type of *contexts* (geographic, climatic, medical, social, etc.) through an automatic entity context binding, knowing that a context can have an impact on another context. An *entity context binding* is like a bridge built at run-time during a context collection process. This binding allows answering to the following question: For a given scenario, which relevant context information do we need to collect? The choice of the context to take into account in a message routing scenario will depend on various factors, such as the content and subject of the message, and the role of the actors in message sending. For instance, in a medical emergency scenario, a bridge needs to be created at run-time between the environmental contexts (hostility, weather, access) of the target and the source, since there is a high probability of persons transfer. Each context corresponds to a multitude of attributes, also called contextual information or observations in context-aware systems. An attribute can be interpreted from a set of basic attributes or not. For instance, climatic conditions depend on different parameters: temperature, snow, etc. In addition, we distinguish two classes of attributes that we call background and real-time attributes. *Background attributes* are the relatively static observations which practically don’t change their values during message transmission (e.g., the job of the message sender). Conversely, *real-time attributes* (e.g., sender geographic localization) are dynamic observations which can take new values at run-time and thus might trigger changes in the routing policy behavior. Background and real-time attributes can be interpreted or not. The *context aware adaptive routing policy* constitutes the core of the proposed model. It clearly shows the dependency, as well as the sensitivity of the routing policy applications to context information stemming from source, target and message ecosystems.

IV. MODEL-DRIVEN ARCHITECTURE FOR CONTEXT-AWARE ROUTING POLICY MANAGEMENT

In this Section, we present our functional architecture (see Figure 2), which aims to be as generic as possible and to comply with different scenarios, with the objective to allow the steering of an intelligent, adaptable and flexible message routing policy. The architecture we propose is composed of five components we describe hereafter and it is mainly based on Service Oriented Architecture.

A. *Message reporter*

This module represents the source or the origin of the message. It is responsible for message reporting and publishing to the context-aware intelligent routing policy manager (RPM). It is usually a device (smartphone, PDA, PC, phone, etc.), controlled by a person or a software application.

B. *Ecosystem supervisor*

The main role of this module is to supervise and to listen continuously to the ecosystems. It compares real-time and background context attributes, detects changes and notifies the RPM. Notifying the RPM is triggered when observations differ from more than a given threshold from their previous values. It is like a contract carried out between the RPM and

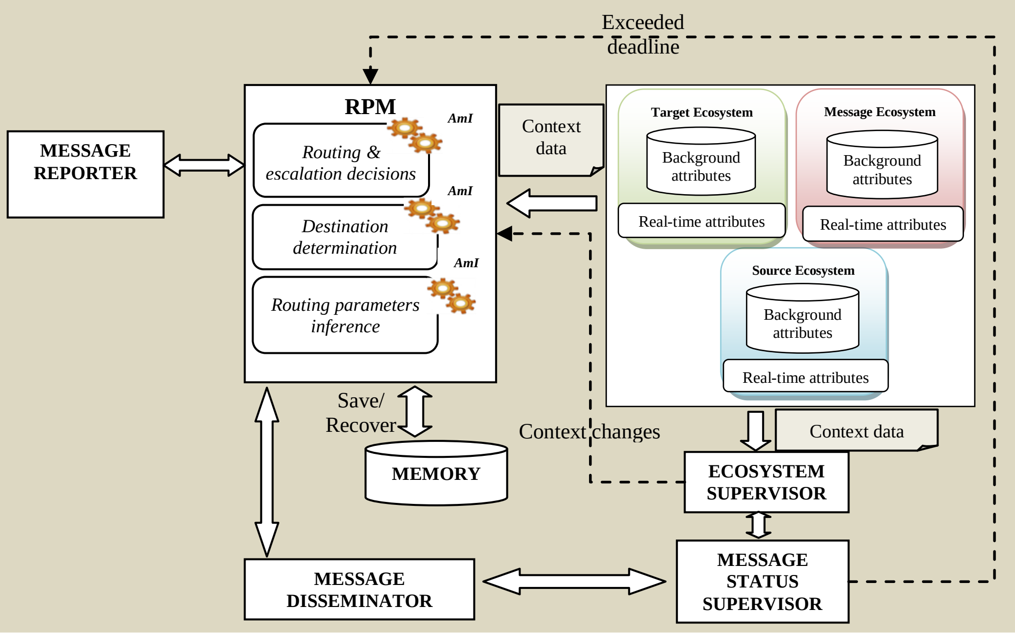


Figure 2. Functional architecture for context-aware routing policy management

the ecosystem supervisor to identify the situations under which the RPM requires behaviors adaptation.

C. *Message status supervisor*

This module supervises the status of the message. It shall detect deadlines exceeding of message reception, reading and reply, and notify the ecosystem supervisor and the RPM, which shall adapt its decision, if necessary. The message status supervisor activates its own timing system each time it receives from the message disseminator a notification indicating a message sending and the required delay for reading, reception and reply. To this aim, we can adopt a Timed Petri Net (TPN). Indeed, TPN is a convenient method to analyze and model the dynamics of self-adaptive systems [13]. Its main advantage comes from its expressiveness of timing constraints, such as durations of performances and deadlines. Several recent works [16] [17] highlighted some of the key features of TPN, such as safety, liveness and reachability. In our case, the message chain history can be designed as a TPN model where place consists of the message state (waiting for reception/ for read/ for reply) and token represents the message itself. The admissible duration of a message stay in a given place corresponds to required delay already determined by the routing policy manager (see Section E). We can take advantage from TPN at two levels: on one hand, it avoids message deadlocks, thus enabling the possibility of message redirection; on the other hand, TPN participates in the determination of the destination’s trust level, which is considered among the most important criteria to take into account when identifying relevant destinations.

D. *Message disseminator*

It receives the message from the RPM and dispatches it to the destinations endpoint devices. It reports also the communication message status to the RPM and notifies the message status supervisor once the message is sent.

E. *Routing policy manager (RPM)*

The RPM constitutes the core of the architecture. It interacts with all the rest of the modules. The RPM is mainly based on ambient intelligence technology, which will allow to reason about the context in order to enhance the quality of message transmission. The module will proceed at operational level and include real-time decisional processes that will react to context, as well as to unpredictable events, such as destination unavailability, localization change, etc.
The RPM has several roles where each one is played by a service component in the RPM structure. The basic responsibilities are as follows:

- Receive messages coming from the message reporter.
- Report the message status to the message reporter.
- Collect contextual information from the source, message and target ecosystems.
- Analyze data to make them acceptable and exploitable for interpretation.
- Infer routing parameters: For each scenario, the RPM determines the requirements and the appropriate parameters essential to route the message to the relevant destination.
- Assign the demand to the appropriate destination.
- Dispatch the message to the message disseminator.
- Cancel message routing by asking the message disseminator for stopping dissemination in order to redirect the message to another destination, if necessary.

- Save/recover contextual data in/from the cache memory.
- Delete messages from the memory in case of request from the message reporter or disseminator (e.g., because of obsolete or undeliverable messages).

The routing policy manager must face two major challenges: (1) to well understand the context information; (2) to dynamically adapt its behavior at run-time according to the context. In our design, the routing policy adaptation will operate at three levels, each of which constitutes an RPM service that we describe hereafter.

1) Routing and escalation decision: At run-time, an escalation (escalate, descalate) and/or a routing decision ((re)send, redirect, cancel) may be triggered/modified, depending on the ecosystem supervisor, message status and message disseminator results notifications.

For instance, the service can change/modify the message itinerary because of changes in the destination context (e.g., geographic localization) or for exceeding the required message reply delay detected by the message status supervisor, or because of connection problems identified by the message disseminator. In case of message redirection, the routing and escalation service calls the routing parameters inference service to determine new requirements, viz a new typical destination profile.

2) Routing parameters inference: The mission of this service is to reason about context information belonging to the message and source ecosystems in order to determine the message routing parameters, already defined in the previous Section. The routing parameters inference service determines, for each scenario, a destination profile type, as well as the required delays for message reception, reading and reply.

For that purpose, classical rule based techniques can be used to infer routing parameters in function of the context information. Such an approach however, although appropriate for static applications, is rather difficult to set up for dynamic applications. Also, building the rules set requires to predict all possible context configurations, which is not so evident.

For solving the routing parameters inference problem, we have two dimensions to take into account: *(1) Time* and *(2) uncertainty*.

a) Time: In some scenarios, at run-time, the message can follow multiple routing policies. The routing parameters and the observations at time t may depend on routing parameters and observations at previous time t-1.

b) Uncertainty: To reason about context information, the routing parameters inference service must be able to deal with uncertainty. Indeed, according to the sources they come from (e.g., noisy sensors), context information can be uncertain, incomplete or imprecise.

Hence, we have to find the appropriate tool that is able to model the routing policy dynamicity in function of time and uncertainty. The objective is to make the routing policy intelligently adaptive according to observations evolving over the time.

Dynamic Bayesian Networks (DBNs) [18] may be suitable for that. A key feature of a DBN is to unify the representation of temporal dimensions and of uncertainty. A DBN is a Bayesian Network which relates variables to each other over adjacent time steps called time slices. These temporal connections incorporate conditional probabilities between variables based on the Markovian condition that the state of the system at time t depends only on its immediate past, i.e., its state at time t-1. Based on the stochastic formalism, DBNs allow to infer the probability of unknown states, given some known observations and the initial probability distributions. Initial probabilities may be computed on the basis of experimental data with machine learning technics. Probabilistic inference is defined as the process of deriving logical conclusions from known, or assumed to be true, premises. The problem of inference in DBN consists in finding P ($X^{t-1}|Y^{t-1}$), where Y^{t-1} represents a set of t consecutive observations, and X^{t-1} is the set of the corresponding hidden variables. Forward-backward and junction tree algorithms are some examples of inference algorithms that may be used in DBNs.

In our case, we can imagine using a DBN to infer the probability of the message routing parameters as hidden variables, based on observations coming from the source and message ecosystems. For instance, a DBN might be used to estimate the probability level of the requested destination staff type (medical, rescue, assistance, etc.) and use it as a routing parameter, given the source ecosystem observations (message expeditor trust level high/medium/low), the message subject importance level (high/medium/low), and the message ecosystem observations (informative/alarm message type, emergency level high/medium/low, etc.).

3) Destination determination: This service addresses the following question: for a given scenario, to which relevant destination the message must be sent? Using a set of preconfigured recipients, the destination determination service shall search for the nearest destination that is closest to the typical destination profile which has already been determined by the routing parameters inference service. This process needs beforehand an in-depth analysis of the target ecosystem context. To select the relevant destination, several methods can be adopted such as multi-criteria utility function or K-Nearest Neighbor algorithm (K-NN) [19].

V. USE CASE: MESSAGE ROUTING ADAPTATION IN HEALTHCARE APPLICATION

In this Section, we illustrate the need for implementing dynamic message routing policy management applications with a scenario stemming from the healthcare domain. Obviously, healthcare applications are both mission-critical and real-time since they require in-time responses especially in emergency cases. The following scenario shows how the RPM will adapt its behaviors according to context changes belonging to the source ecosystem.

Scenario: Patient A has a history of cardiac disease. He visits a high mountains area for skiing, taking his intelligent cardiac device with him. While skiing, he felt a chest pain. His care device reports an alarm message to the RPM. In this case, the care device operates as a message reporter. The latter sends the message to the RPM which confirms the message reception. The RPM collects as a first step interpreted and/or non-interpreted, background and/or real-time contextual data from the source ecosystem (e.g., interpreted real-time attribute: environment type: hostile; interpreted background attribute: history cardiac problem: yes) and from the message ecosystem (e.g., interpreted real-time attribute: message importance level). The RPM unifies the collected observations and calculates the initial probability distribution necessary for the DBN. The latter infers a typical profile satisfying such criteria, e.g., staff type: medical, the required resources materials (viz an helicopter because of hostile patient A environment) and the delays for message reception, reading and reply. In a preconfigured destination list, the RPM searches a profile that is nearest to the typical profile and associates the message to patient A's admitting physician (physician B). The message disseminator sends the message to physician's B PDA. Physician B confirms message reception within the specified delay; however he exceeds the delay required for replying. In between, the patient' chest pain has become more acute. The ecosystem and message status supervisors notify the RPM of these changes. New observations are notified to the RMP which adapts its decisions by escalating the message priority and redirecting it to an emergency department as a new destination, which thus takes care of the patient. Let us note that a history of the exceeded deadlines and negative responses may decrease physician B's trust level which is considered an important criteria to take into account when choosing the relevant destination. Indeed, some specific context situations and reasons for which a physician may reject a healthcare request shall also be taken into account. For example, physician B can be unavailable because of commitment in another task, of vacations, etc.

VI. CONCLUSION

In this paper, we present a model and an architecture that emphasize the multidimensionality of the message routing problematic. Our objective is to ensure QoS of message transmission in terms of delivering the message to the right destination with respect to the required delays for message reception, reading and reply, and taking into account message, source and target context changes. Meanwhile, we believe that developing an intelligent adaptive routing policy in pervasive environments may save lives and money, especially in emergency scenarios. Within the proposed architecture, we also highlighted three adaptive services for which we propose appropriate methods to make them sensitive to context changes and to exceeded message deadlines. One of the major assets of our data and model driven architecture approach is to embed artificial intelligence methods at different levels: to infer routing parameters, to choose the right destination, and to select relevant context data thus avoiding the need for big data exchange.

ACKNOWLEDGMENT

This research work was supported by grants from AXON' SAS F-51210 MONTMIRAIL France and ANRT.

REFERENCES

[1] A. Saeed and T. Waheed, "An extensive survey of context-aware middleware architectures," IEEE International Conference on Electro/Information Technology (EIT 2010), IEEE Press, May 2010, pp. 1–6, doi: 10.1109/EIT.2010.5612118.

[2] M. Salehie and L. Tahvildari, "Self-adaptive Software: Landscape and Research Challenges," ACM Transactions on Autonomous and Adaptive Systems, vol. 4, May 2009, pp. 14:1–14:42, doi: 10.1145/1516533.1516538.

[3] Y.-F. R. Chen and C. Petrie, "Guest editor's introduction - Ubiquitous mobile computing," IEEE Internet Computing, vol. 7, Mar. 2003, pp. 16–17, doi: 10.1109/MIC.2003.1189184.

[4] C. Taconet, Z. Kazi-Aoul, M. Zaier, and D. Conan, "CA3M: A Runtime Model and a Middleware for Dynamic Context Management," in On the Move to Meaningful Internet Systems, vol. 5870, R. Meersman, T. Dillon, P. Herrero, Eds. Heidelberg, 2009, pp. 513–530.

[5] C. Ramos, J. C. Augusto, and D. Shapiro, "Ambient Intelligence-the Next Step for Artificial Intelligence," IEEE Intelligent Systems, vol. 23, Mar. 2008, pp. 15–18.

[6] Y. Sun, W. Tan, L. Li, G. Lu, and A. Tang, "SLA detective control model for workflow composition of cloud services," IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2013), IEEE Press, Jun. 2013, pp. 165–171, doi: 10.1109/CSCWD.2013.6580957.

[7] R. Ding and G.-M. Muntean, "A context-aware cross-layer energy-efficient adaptive routing algorithm for WLAN communications," IEEE 37th Conference on Local Computer Networks (LCN 2012), IEEE Press, Oct. 2012, pp. 176–179, doi:10.1109/LCN.2012.6423600.

[8] S. Peng, S. X. Yang, S. Gregori, and F. Tian, "An adaptive QoS and energy-aware routing algorithm for wireless sensor networks," International Conference on Information and Automation (ICIA 2008), IEEE Press, Jun. 2008, pp. 578–583, doi: 10.1109/ICINFA.2008.4608066.

[9] F.-S. Hsieh, "Context-aware Workflow Driven Resource Allocation for e-Healthcare," 9th International Conference on e-Health Networking, Application and Services (HealthCom 2007), IEEE Press, Jun. 2007, pp. 34–39, doi: 10.1109/HEALTH.2007.381599.

[10] N. B. Behlouli, C. Taconet, and G. Bernard, "An architecture for supporting Development and Execution of Context-Aware Component applications," ACS/IEEE International Conference on Pervasive Services (ICPS 2006), IEEE Press, Jun. 2006, pp. 57–66, doi: 10.1109/PERSER.2006.1652207.

[11] Y. Oh, J. Han, and W. Woo, "A context management architecture for large-scale smart environments," IEEE Communications Magazine, IEEE Press, vol. 48, Mar. 2010, pp. 118–126, doi: 10.1109/MCOM.2010.5434383.

[12] M. Miraoui, C. Tadj, J. Fattahi, and C. Ben Amar, "Dynamic Context-Aware and Limited Resources-Aware Service Adaptation for Pervasive Computing," Advances in Software Engineering, vol. 2011, Feb. 2012, pp. 1-11.

[13] N. Cardozo, S. Gonzalez, K. Mens, R. Van Der Straeten, and T. DHondt, "Modeling and Analyzing Self-Adaptive Systems with Context Petri Nets," International Symp. on Theoretical Aspects of Software Engineering (TASE 2013), IEEE Press, Jul. 2013, pp. 191–198, doi: 10.1109/TASE.2013.33.

[14] F. Fleurey, V. Dehlen, N. Bencomo, B. Morin, and J.-M. Jézéquel, "Modeling and Validating Dynamic Adaptation," in Models in Software Engineering, vol. 5421, M. R. V. Chaudron, Eds. Heidelberg, 2009, pp. 97– 108.

[15] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," Proc.1999 Handheld and

Ubiquitous Computing Symp. (HUC 1999), Springer-Verlag, 1999, pp. 304–307.

[16] Z. Hu, T. Lu, and Z. Zhao, “Context-aware service system modeling using timed CPN,” 10th International Conference on Service Systems and Service Management (ICSSSM 2013), IEEE Press, Jul. 2013, pp. 164–169, doi: 10.1109/ICSSSM.2013.6602579

[17] R. B. Dilmaghani and R. R. Rao, “Supervisory decision making in emergency response application,” IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2013), IEEE Press, Mar. 2013, pp. 553–558, 10.1109/PerComW.2013.6529557.

[18] Z. Ghahramani, “Learning dynamic Bayesian networks,” in Adaptive Processing of Sequences and Data Structures, vol. 1387, C. Lee Giles, M. Gori, Eds. Heidelberg, 1998, pp. 168–197.

[19] G. Gutin, A. Yeo, and A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP,” Discrete Applied Mathematics, vol. 117, Mar. 2002, pp. 81–86.

CObAPAS: Combinatorial Optimization based Approach for Autonomic Systems

Pedro F. do Prado, Luis Nakamura, Marcos Santana, Regina Santana
Institute of Mathematics and Computer Science - University of São Paulo
São Carlos, SP, Brazil
Email: {pfprado,nakamura,mjs,rcs}@icmc.usp.br

Omar A. C. Cortes
Federal Institute of Maranhão
São Luis, MA, Brazil
Email: omar@ifma.edu.br

Abstract—**This paper proposes a new approach to develop autonomic systems or transform traditional systems into autonomic ones. This approach is based on defining the autonomic module of the system as a combinatorial optimization problem. After that, a wide range of different techniques can be used to implement the autonomic module of the system. This study addresses two major problems: autonomic system specification and autonomic system evaluation. The former helps the developer to understand the system goals, constraints and scope, the latter, helps the developer quantitatively evaluate the efficiency of different techniques of implementing the autonomic module of the system. A case study demonstrates the viability and effectiveness of the proposed approach.**

***Keywords*–Autonomic systems; combinatorial optimization based approach for autonomic systems; QoS-aware service selection; combinatorial optimization problems; performance evaluation.**

I. Introduction

The concept of self-adaptation is presented in many research areas like: biology, chemistry, logistics, etc.. Self-adaptivity in computer-based systems is relatively newer. Some of the first references to self-adaptive computer systems are from the late 1990s. The term self-adaptation covers multiple aspects of how a system reacts: Self-Awareness, Context-Awareness, Self-Configuring, Self-Optimizing, Self-Healing and Self-Protecting. There are two approaches for creating self-adaptive systems: centralized and decentralized. In the centralized one, the analysis and planning are concentrated in one single entity. Furthermore, this form of self-adaptation has the advantage of cohesiveness and low communication overhead if compared with a decentralized mechanism [1]. An **Autonomic System** (AS) is an example of centralized self-adaptive system. On the other hand, decentralized self-adaptation, distributes the analysis, planning, or the feedback mechanism among different parts of the self-adaptive system. **Autonomic computing** (AC) is the computing paradigm behind an AS. The general idea is to mimic the autonomous nervous system of humans, which concentrate itself on higher-level objectives, instead of more specific and detailed aspects. For example, a person can concentrate on writing a letter instead of actively controlling the heartbeat, blood pressure, level of insulin on the blood and so on.

AC constitutes an important computing paradigm to automate complex systems management and reduce the need of human intervention. It can be applied to modern and widely used commercial solutions. One of the most used Cloud Computing services provider, the Amazon Elastic Computing Cloud (EC2), provides some tools for self-managing the users systems, by means of increasing or decreasing the number of Virtual Machines (VMs), according to the users demand and previous defined policies. Companies like: Netflix and the Jet Propulsion Laboratory/NASA uses EC2 solution [2].

In [3] Affonso et al. proposed a reference architecture for self-adaptive software. They present an adapted control loop based on **M**onitor, **A**nalyze, **P**lan and **E**xecute, based on **K**nowledge (MAPE-K) and define the modules that must be implemented in order to achieve this reference model. However, they are focusing on how to solve a problem (how to implement an autonomic control loop) and not on how to define the problem that must be solved by this autonomic control loop. The authors in [4] proposed a benchmarking framework for distributed autonomic systems. They also do not focus on how to define the problem that the autonomic control loop must solve. In other cases, frameworks were proposed to help the development of autonomic systems. Although, these frameworks are useful, they usually focus on some specific paradigm or architecture, i.e., Service-Oriented Architecture (SOA), sensor networks or cloud computing [5][6]. Other related works focused on creating detailed and domain-specific performance models of systems, using queuing network models or Petri nets that can be used by an AM to implement aspects like self-configuring and self-optimization. These models are mostly domain-specific, complex to create and validate, and cannot be easily adapted in cases of changes in the system [7][8]. Further, it is important to point out that some related works give qualitative and general information about the performance of different techniques to implement an AM, like Artificial Neural Networks (ANN), linear feedback control, performance model based adaptive control, decision tree and so on [9][10]. Finally, most of the related works deal with one or two aspects of AC, like self-healing and/or self-configuring [5][7]. In this paper, we focus on developing an approach to define the problem that must be solved by an autonomic system. Firstly, we present the steps required to define the problem that an autonomic system must solve. Secondly, we present a simple case study to demonstrate the viability of the proposed approach. Finally, we provide some ideas to future works that can improve the proposed approach and other possible applications.

We chose the domain of **QoS-aware Service Selection** (QSS) to develop our case study. This domain is suitable for AC because its environment is highly dynamic and must be able to deal with changes in workload, QoS preferences, fault tolerance and so on. We transformed a traditional QSS system into a self-configuring and self-optimizing one to demonstrate the viability of CObAPAS.

This paper is organized as follows: in Section II the concepts related to AC and QoS-aware service selection are presented. Section III contains the approach to develop AS. Section IV presents a case study to validate the proposed approach. Finally, in Section V are presented the conclusions and future work.

II. Autonomic computing and QoS-aware service selection

A. Autonomic computing

The automation of computational resources management is not a new problem for computer scientists. For decades, software components have evolved to deal with the growing complexity of performing the control of systems, sharing resources and execution of operational management [11]. **Autonomic computing** is a computational paradigm based on biological systems that aim to deal with the management of complex systems, offering the possibility of self-management minimizing the need for human intervention [12]. Autonomic computing is based on four principal attributes, namely [11]:

- **Self-configuring:** dynamically configure itself, a system can adapt (with minimal intervention) to the deployment of new components or changes in the system.
- **Self-healing:** detect problematic operations and then initiate corrective actions without disrupting system applications.
- **Self-optimizing:** efficiently maximize resource allocation and usage to meet end users' needs with minimal intervention. It addresses the complexity of managing system performance.
- **Self-protection:** detect hostile or intrusive behavior as it occurs and take autonomous actions to make itself less vulnerable to unauthorized access and use, viruses, denial-of-services attack, and general failures.

Autonomic systems are composed of two parts: Autonomic Element(s) (AE) and Autonomic Manager (AM). An AE can be divided into: hardware (computers, printers, routers, etc.) and software (web service, application container, virtual machine, etc.). The communication between AM and AEs occurs using **Sensors** and **Effectors**. Sensors collect data about the AEs. On the other hand, Effectors have the function of performing the operations sent by the AM to the AEs. The AM implements a control loop based on four activities: **M**onitor, **A**nalyze, **P**lan and **E**xecute, based on **K**nowledge (**MAPE-K**).

Monitor: monitors and collects the relevant details of interest from the managed element. **Analyze:** analyzes information provided by the monitor activity to determine if it is necessary to take some action. If some action is required, it is passed to the plan activity. **Plan:** creates a plan (or a sequence of actions) by structuring actions to achieve system goals. **Execute:** performs the actual actions, hence changing the behavior of the managed element. The **K**nowledge Base contains information about the system, that must be monitored, the different available plans and so on [13].

B. QoS-aware service selection

Quality of Service (QoS) is a set of non-functional properties of Web services. Some of well known QoS attributes are: cost, response time, availability, security, and so on. QoS-aware web services composition (QWSC) is defined as an integration of different services aiming to attend complex business needs. For example, instead of manually accessing a service for buying an airplane ticket, and after that another service to reserve a hotel room, the user can access a composed service that performs both tasks. QWSC is divided into two parts: creation of the composition flow and QoS-aware service selection. In the former, the developer of the composed web service can use some business process modeling language, like Web Services Business Process and Execution Language (WS-BPEL). Using WS-BPEL the developer will define the order of execution of the services, the exchange of data between them and if some services will execute in sequential or parallel order [14]. Figure 1 shows the division between these activities.

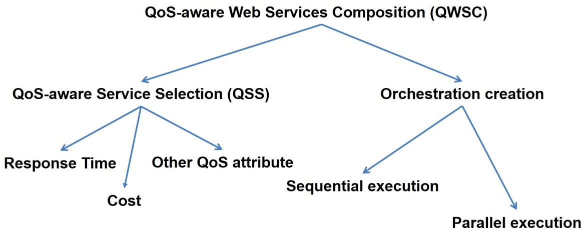


Figure 1. Different aspects of QoS-aware Web Services Composition.

QoS-aware service selection is based on QoS attributes of services. It means that based on the QoS attributes the algorithm or other technique of service selection will decide which service will be included on the composite service. There are a wide range of different techniques to store and retrieve information about QoS attributes of services. They can be stored and retrieved in a relational database application [15] or using some semantic parallel approach [16]. QoS-aware service selection is a combinatorial optimization problem and is NP-Hard, thereby, many related works spent efforts developing and testing algorithms to solve it.

III. Development of autonomic systems based on combinatorial optimization problems

A. Motivation

This section will show our proposed approach to develop new autonomic systems or transform traditional systems into autonomic ones. The main idea of this approach is to provide a clear and easy form that can be used in a wide range of systems. The proposed approach is named CObAPAS: Combinatorial Optimization based Approach for Autonomic Systems. We define the problem that the AM must solve as a combinatorial optimization problem. CObAPAS has the following advantages:

- It is independent of architecture and/or technologies: it can be used from a simple web server to complex cloud environments.
- It can be used to create new autonomic systems or transform traditional systems into autonomic ones.
- It provides a formal and clear definition of the problem that must be solved.
- It allows to evaluate different solutions proposed for the stated problem quantitatively.
- It can address one or more aspects of AC, it can be constrained or unconstrained and it allows even

multiple constraints and/or multiple objectives to be minimized or maximized.

The required steps of CObAPAS are: **system definition**, **search-space definition**, **objective function definition** and **developing solutions, test scenarios and evaluation**. They will be shown in the next subsections. In Section IV, we present a case study to help illustrate our approach.

B. System definition

First, we must decide which system we aim to create or modify. For example, we can develop a web server from scratch or transform a traditional web server into an autonomic web server. The system could be composed of one single entity, i.e., a web server or can be composed of two or more entities, i.e., a system composed of a web server and a database application. In fact, a system could be very simple or composed of many entities that interacts with each other in different manners. Once we have defined the system, we can continue with the next steps.

C. Search-space definition

After choosing the system, it is necessary to define the search-space of our problem. The search-space is the set of attributes that should be modified to optimize the objective function that will be created on the next step. The search-space size is the number of all possible combinations of all defined attributes. For example, if we have ten attributes and each one can assume two values, the search-space size will be 2^{10}. It varies according to the system, and **the only restriction is that they all must be discrete**. Since we are dealing with combinatorial (or discrete) optimization problems, all attributes must be discrete. We have to define which attributes we want to consider in our system; in a system composed of a web server and a database application for example, there are some parameters that can be dynamically modified in execution time. Therefore, we can define that some of these parameters are our search-space and include them into our problem definition. Examples of such parameters are shows in Table I.

TABLE I. LIST OF PARAMETERS.

Web Server (IIS 5.0)	**Database Server (SQL Server 7.0)**
HTTP Keep Alive	Cursor Threshold
Connection Timeout	Locks
MemCacheSize	Priority Boost
MaxPoolThreads	Max Server Memory

D. Objective function definition

Now, we must decide which aspect(s) of AC we want to focus on, and other characteristics, such as if the problem will be single-objective or multi-objective, if it will be constrainted or unconstrainted and if the objective function must be minimized or maximized. Once we are dealing with the problem definition, there are no technological or architectural restrictions.

In [17], the authors present a wide range of combinatorial optimization problems, how to define them and some algorithms to solve them. In fact, this study did not focus on solutions for AS, but in the aspect of the problem formalization. In our point of view, any aspect of AC can be defined as an optimization problem. For example, suppose that it is required to develop a QoS-aware service selector (QSS) with Self-Healing capabilities. If some service is unavailable at execution time, the QSS should select an equal or similar service and execute it, instead of that one which is unavailable. We want that in all occurrences of unavailability, the QSS select other service as fast as possible. So, it can be defined as the minimization of average recovery time (time to select a new service and execute it) of the QSS.

Doing so, we can develop two or more solutions for the problem and quantitatively compare them. Therefore, after we have defined some test cases, instead of qualitative and generic affirmations, we can quantitatively compare the proposed solutions. In fact, this approach can be used to define the problem according to the AS developer's needs.

E. Developing the solution(s), creating test scenarios and evaluating the solutions

After we have defined the problem, we need to develop solutions for it. It is possible to use from simple static policies to heuristic algorithms or even complex and detailed queuing network models. Since the problem is formally defined, if we have two or more solutions, they can be quantitatively compared.

In order to achive an effective and a properly evaluation of the proposed solutions, it is mandatory to define some experiments which reflect possible real scenarios that the AS will face with. The authors in [18] explains in many details how to define a set of experiments, workloads, how to use statistical tools and so on.

After all these steps, the AS system is formally defined, with its solutions quantitatively compared. If more solutions arise, they can also be compared with the old ones. If something change after some period (for example, a new constraint must be added to the objective function), the objective function must be updated and the solutions must be re-evaluated.

IV. Self-configuring and self-optimizing QoS-aware service selector: A case study

A. Motivation

Developing large-scale distributed systems presents the challenge of providing a way for software to adapt to changes in a computational environment. In response, the system must be able to handle all changes in the workload, failures, changes in QoS preferences, and so forth [19]. Furthermore, the need of developing systems that are capable of self-adapting is becoming greater [20].

The context of QoS-aware service selection is highly dynamic and susceptive to changes. For that reason, it is recommend that the system responsible for the service selection should be autonomic, instead of manually controlled by humans [21][22]. For example, if a service provider is overloaded, the average response time of its services can be unsatisfactory, so it should not be selected until its average response time returns to an acceptable level.

B. Problem definition

This case study will be as simple as possible, with the objective of showing how following the steps mentioned in Section III can lead to a well-defined combinatorial optimization problem, which helps to change a traditional system into an autonomic one. The selected system is a QoS-aware web service selector, proposed by the authors in [15]. Five different algorithms were implemented and a performance evaluation was made. The QoS attributes considered were: availability, cost, response time, reputation and confidentiality.

Considering that each Web Service has its own QoS attributes, it is necessary to use aggregate functions for computing the QoS of the composition plan as a whole [15]. For example, Table II, described in [15], shows an example of aggregation of these attributes:

TABLE II. QUALITY OF SERVICE ATTRIBUTES.

Availability	$\prod_{i=1}^{i=n} availability(WSi)$
Cost	$\sum_{i=1}^{i=n} cost(WSi)$
Response Time	$\sum_{i=1}^{i=n} responseTime(WSi)$
Reputation	$\sum_{i=1}^{i=n} reputation(WSi) * 1/n$
Confidentiality	$\sum_{i=1}^{i=n} confidentiality(WSi) * 1/n$

The Web Services composition plan could be described as a sequence of tasks (abstract Web Services) with an initial and a final task. For any abstract WS, it could have some candidate services (concrete Web Services) with same or similar functionality but different QoS attributes. Thus, there are various composition plans for each execution path of composite service. For example, if there is one execution path, with 10 abstract WS and 15 (concrete Web Services) per abstract Web Service, then the number of composition plans should be about 15^{10} [15]. Table II presents the aggregate functions of QoS attributes considered in this paper. However, it is also necessary a form to assess the QoS of the composition as a whole, taking into account the QoS attributes defined. The function to be **maximized** in the experiments is shown in (1), considering A (**Availability**), C (**Cost**), RT (**Response Time**), R (**Reputation**) and Con (**Confidentiality**).

$$F(x) = A + C + RT + R + Con \tag{1}$$

Given that the QoS attributes were normalized in a form that 0 is the worst result and 1 is the best result possible, simply add up all the attributes of QoS, regardless if they have to be either minimized or maximized. The Equation 2 and Equation 3, presented in [23], represents respectively, the equation used for attributes that must be minimized and the equation used for the attributes that must be maximized. Then, for each QoS attribute, the aggregated QoS is calculated using the formulas presented in Table II. Thereafter, the composition aggregated QoS is computed using the formula shown in (2). Finally, this number is normalized between 0 and 1 and called Normalized Composition Aggregated QoS (NCAQ).

By doing that, we already accomplished **step one** and defined the system. The **step two** is to define the search-space. In this case study, only one attribute will be considered: static policy of the system. Static policies can be any fixed rule or algorithm that is used to implement the AM of the system. For example, a static policy can define that an autonomic web server must decline any request if its capacity is above 90%. The static policies are two algorithms developed in [15] and they will be explained in the subsection named Implemented algorithms.

In **step three**, we must define the objective function. In order to define that function,we must consider which aspect(s) of the AC in the system we want to focus on. In our case study, we chose **self-configuring** and **self-optimizing**. After that, we must define if the objective function will be single-objective or multi-objective. If we choose multi-objective, it is necessary to guarantee that two or more objectives are in conflict, otherwise the global solution would be a single point in the search space. For instance, those functions can be something such as minimize the average response time and maximize the average QoS obtained. In our case study, we defined that the function would be **single-objective** and we must **minimize** the **average response time** of the attended requests. Finally, we must define if the objective function would be constrained or unconstrained, we chose **unconstrained**. So, the defined objective function is shown in (2):

$$Minimize \frac{\sum_{i=1}^{i=n} ResponseTime(R_i)}{n} \tag{2}$$

where *n* is the number of requests and $ResponseTime(R_i)$ is the response time of request R_i.

A wide range of different techniques can be used: ANN, heuristic algorithms, static policies, adaptive performance models and so on. Since in this paper we do not focus on the solutions, we chose static policies.

Step four is divided into three phases: developing the solution(s), creating test scenarios and evaluating the solutions. The solutions used in this experiments are described in subsection Implemented algorithms. The test scenario is described in the subsection Experiment design and the evaluation of the solutions is described in subsection Result analysis.

C. Implemented algorithms

Exhaustive Search (ES): This algorithm, also known as “brute force”, analyses all points in the search space. In the case of the QWSC problem, it compares the QoS obtained by all possible combinations of composite plans and returns the best one (with higher QoS). So, the obviously advantage of this algorithm is that the global optima are always guaranteed. The disadvantage is related to their computational complexity, because it is exponential. For instance, suppose a composite flow has ten abstract WS and one hundred concrete Web Services per abstract Web Service, the number of points in the search space will be 100^{10}, which will probably take hundreds of years to be calculated. Because of that, this algorithm could be used only in small search-space sizes, because of the soft real-time characteristic of the QWSC problem.

Greedy Heuristic (GH): This algorithm was an original idea proposed by the authors in [14]. For each abstract WS in the composite flow, the algorithm evaluates all concrete Web Services available for that abstract WS and selects the one with higher aggregate QoS. Due to all QoS attributes are normalized between 0 and 1 (and the highest is always the best one), it is necessary to calculate the sum of all QoS attributes of all concrete Web Services. The one with higher aggregate QoS is selected to its respective abstract WS. Suppose *j* is the current WS to be evaluated, *k* is the number of QoS attributes and *q* is the current QoS attribute, (3) represents the algorithm:

$$GH(WSj) = \sum_{i=1}^{i=k} q_i \tag{3}$$

The advantage of this algorithm is that it is very fast because it is directly related to the number of total concrete Web Services, i.e., suppose a composite flow with four abstract Web Services and one hundred concrete Web Services per abstract Web Service, the number of total concrete Web Services will be four hundred. So, the algorithm should calculate the aggregate QoS function of four hundred concrete Web Services; instead of calculating 100^4 composite plans like the ES algorithm does. The disadvantage of this algorithm is that it could not benefit from a larger deadline, because it is a deterministic algorithm.

D. Experiment design

The main goal of this study is to evaluate different policies to solve (2). Thus, the test environment is composed of three machines: one representing a client, another a service provider and a third one executes a MySQL server with the data about the QoS attributes of the Web services. In the considered environment, the three machines are in the same network and are linked by a gigabit network switch. The machines used are heterogeneous and their configuration is presented in Table III.

The experiments were conducted varying three factors in order to verify the performance of the policies and different number of abstract Web Services and concrete Web Services per abstract Web Service. The parameterization of these factors can be observed in Table IV. All experiments were executed ten times and the average response time was colected and presented in Figure 2.

TABLE III. ENVIRONMENT CONFIGURATION.

Machine	CPU	Clock	Cache	RAM
Service provider	Intel® Core™2 Quad	2.66 GHz	3 MB	8 GB
MySQL server	Intel® Core™i3	3.10 GHz	3 MB	4 GB
Client	Intel® Core™2 Quad	2.4 GHz	4 MB	4 GB

TABLE IV. LIST OF EXPERIMENTS.

Exp. number	abstract WS	concrete WS	Algorithm
1	2	100	ES
2	2	200	ES
3	3	100	ES
4	3	200	ES
5	2	100	GH
6	2	200	GH
7	3	100	GH
8	3	200	GH

E. Result analysis

The objective of these experiments was to discover which policy is most effective in optimizing the defined objective function. For this purpose, eight experiments were conducted, varying the number of abstract Web services and the number of concrete Web services for each abstract Web service.

In all experiments, the GH policy was more effective, since the average response time was considerably lower. In some cases, the average response time of the ES policy was more than two times the GH average response time. The lines inside the columns represents the calculated confidence interval (CI) (it was defined a 95% degree of confidence) and it is related to the variability of the results. In all experiments, the CI was lower in the GH policy. Considering that, GH is not just faster but also more stable than ES.

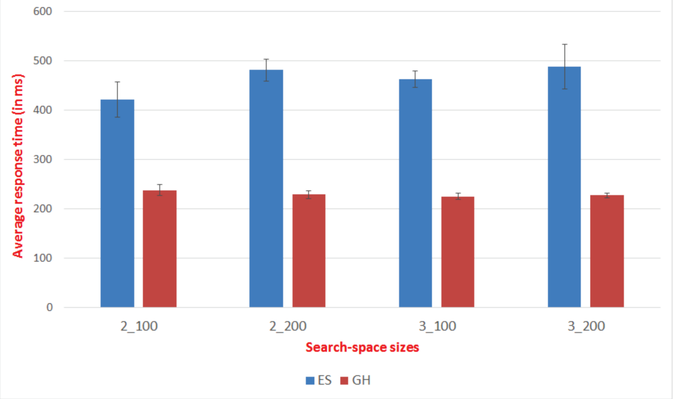


Figure 2. Average response times of ES (blue) and GH (red) in milliseconds.

V. Conclusions and Future Work

This paper presented CObAPAS, a new approach to develop new autonomic systems or transform traditional systems into autonomic ones. It was discussed the importance of autonomic computing and the motivation for developing an approach that helps the problem formalization of an AS and has not architectural and/or technological limitations.

Compared to related works, our paper focuses on the problem formalization instead of proposing solutions for specific autonomic systems. Our approach can fit into the AS developer's needs since all attributes in the defined search-space are discrete.

A simple study case was presented, to validate our approach. In fact, we believe that many different AS can be created using CObAPAS. The experiments showed that it is possible to quantitatively compare different solutions for the AM, after the objective function was defined. CObAPAS provides two major benefits: guidelines for developing an AS and a way to quantitatively measure the quality of different solutions for the defined problem.

In future works, we plan to develop more sophisticated case studies to validate our approach, with multiple aspects of AC and/or multiple constraints. One example of case study is an autonomic Virtual Machines (VMs) manager. We will use the Famav tool, presented in [24]. Famav is a command line tool for managing VMs. Compared to Virsh (another command line tool) Famav presents a lower performance, but its ease and practicality minimizes this difference. We also plan to create another two approaches to develop new AS or transforming traditional systems into autonomic ones: one for AS systems based on continuous optimization problems and one for systems based on both continuous and combinatorial optimization problems. These new approaches also need some case studies to be validated and to show some applications in real-world problems.

ACKNOWLEDGMENT

The authors would like to thank the financial support of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and CAPES (Coordenação de Pessoal de Nível Superior).

REFERENCES

[1] V. Nallur and R. Bahsoon, "A decentralized self-adaptation mechanism for service-based applications in the cloud," IEEE Transactions on Software Engineering, vol. 39, no. 5, 2013, pp. 591 – 612.

[2] Amazon elastic compute cloud (amazon ec2). website. [Online]. Available: http://aws.amazon.com/ec2/ [retrieved: April, 2015]

[3] F. J. Affonso and E. Y. Nakagawa, "A reference architecture based on reflection for self-adaptive software," in VII Brazilian Symposium on Software Components, Architectures and Reuse, 2013, pp. 129 – 138.

[4] A. Vilenica and W. Lamersdorf, "Benchmarking and evaluation support for self-adaptive distributed systems," in Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, 2012, pp. 20 – 27.

[5] W. Li, P. Zhang, and Z. Yang, "A framework for self-healing service compositions in cloud computing environments," in IEEE 19th International Conference on Web Services (ICWS), 2012, pp. 690 – 691.

[6] K. Zielinski, T. Szydlo, R. Szymacha, J. Kosinski, J. Kosinska, and M. Jarzab, "Adaptive soa solution stack," IEEE Transactions on Services Computing, vol. 5, no. 2, 2012, pp. 149 – 163.

[7] D. Menascé, D. Barbará, and R. Dodge, "Preserving qos of e-commerce sites through self-tuning: A performance model approach," in ACM Conference on e-commerce, 2001, pp. 1 – 11.

[8] J. M. Ewing and D. A. Menascé, "Business-oriented autonomic load balancing for multitiered web sites," in IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009, pp. 1 – 10.

[9] L. Shen, J. Wang, K. Wang, and H. Zhang, "The design of intelligent security defensive software based on autonomic computing," in Second International Conference on Intelligent Computation Technology and Automation, 2009, pp. 489 – 491.

[10] L. Checiu, B. Solomon, D. Ionescu, M. Litoui, and G. Iszlai, "Observability and controllability of autonomic computing systems for composed web services," in IEEE International Symposium on Applied Computational Intelligence and Informatics, 2011, pp. 269 – 274.

[11] S. O. Schimidt, P. F. do Prado, and A. Silva, Fundamentals of Informations Systems - Fundamentos de Sistemas de Informação. Campus Elsevier, 2014, ch. IT infrastructure and emerging technologies - Infraestrutura de TI e tecnologias emergentes, pp. 77 – 91.

[12] A. Khalid, M. Haye, M. Khan, and S. Shamail, "Survey of frameworks, architectures and techniques in autonomic computing," in Fifth International Conference on Autonomous and Autonomic Systems (ICAS), 2009, pp. 220 – 225.

[13] P. T. Endo, M. S. Batista, G. E. Gonalves, M. Rodrigues, D. Sadok, J. Kelner, A. Sefidcon, and F. Wuhib, "Self-organizing strategies for resource management in cloud computing: state-of-the-art and challenges," in verificar, 2013, pp. 13 – 18.

[14] P. F. do Prado, L. H. V. Nakamura, J. Estrella, M. Santana, and R. Santana, "Different approaches for qos-aware web services composition focused on e-commerce systems," in 13th Symposium on Computing Systems, 2012, pp. 179 – 186.

[15] P. F. do Prado, L. Nakamura, J. Estrella, M. Santana, and R. Santana, "A performance evaluation study for qos-aware web services composition using heuristic algorithms," in The Seventh International Conference on Digital Society (ICDS), 2013, pp. 53 – 58.

[16] L. H. V. Nakamura, P. F. do Prado, R. Libardi, L. Nunes, J. Estrella, R. Santana, M. Santana, and S. Reiff-Marganiec, "Fast selection ofweb services with qos using a distributed parallel semantic approach," in IEEE International Conference on Web Services, 2014, pp. 680 – 681.

[17] H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems, Springer, Ed. Springer, 2004.

[18] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Wiley-Interscience, Ed. Wiley-Interscience, 1991.

[19] D. A. Menascé, H. Gomma, S. Malek, and J. P. Sousa, "Sassy: A framework for self-architecting service-oriented systems," The Journal of IEEE Software, 2011, pp. 78 – 85.

[20] A. J. Ramirez, D. B. Knoester, B. H. C. Cheng, and P. K. Mckinley, "Applying genetic algorithms to decision making in autonomic computing systems," in ACM International Conference on Autonomic Computing, 2009, pp. 97 – 106.

[21] A. Charfi, T. Dinkelaker, and M. Mezini, "A plug-in architecture for self-adaptative web service compositions," in IEEE International Conference on Web Services (ICWS), 2009, pp. 35 – 42.

[22] G. H. Alferez, V. Pelechano, R. Mazo, C. Salinesi, and D. Diaz, "Dynamic adaptation of service compositions with variability models," The Journal of Systems and Software, vol. 91, 2013, pp. 1 – 24.

[23] P. F. do Prado, "Desenvolvimento e avaliação de algoritmos para composição dinamica de web services baseada em qos," Master's thesis, Universidade de São Paulo (USP), 2012.

[24] Y. Neves, L. H. V. Nakamura, P. F. do Prado, and M. Santana, "Famav: Analise comparativa entre ferramentas de gerenciamento de maquinas virtuais," in Proceedings of XV Simposio em Sistemas Computacionais (WSCAD-WIC)., 2014, pp. 1 – 6.

Self-organized Architecture for Sharing Data Streams at Large Scale

Nicolás Hidalgo and Erika Rosas

Department of Informatics
Universidad de Santiago
Santiago, Chile
Email: name.lastname@usach.cl

***Abstract*—Stream Processing Engines are designed to deal with real-time computing of massive data streams generated on social networks, news feeding, satellite images, sensor devices, among other sources. For example, in the context of the Internet of Things and Smart Cities, a high volume of data it is expected to be distributed geographically. In this context, the re-use of processed stream enables resource optimization by avoiding re-computation, enabling to provide aggregation and global data visualization. We propose a self-organized architecture to share data streams, which enables resource localization over a scalable, fault-tolerant Distributed Hash Table structure. The Stream Processing Engines are organized into a structured peer-to-peer network and they exploit a Publish/Subscribe system to publish and locate preprocessed streams, possibly in other geographic regions. In order to deal with communication latency problems in the peer-to-peer network, we propose a latency-aware algorithm that estimates distance between the nodes in the system.**

Keywords–Stream Processing; Peer-to-Peer networks; Large Scale Computing; Publish/Subscribe.

I. Introduction

Large scale streams can be generated in domains like meteorology, finance transactions, remote sensing, software logs, wireless sensor network, social interactions, telecommunications, just to mention a few. In order to process the amount of data generated in these scenarios, the capacity of many machines is required.

In the domain of the Internet of Things (IoT) or Smart City platforms, Wireless Sensor Networks (WSN) are used to monitor gas leaks, parking availability, traffic congestion, pollution levels, the infrastructure's health, and garbage levels [1]. These platforms enable integrating and visualizing data in order to make informed management decisions. The technological improvements and lower costs of these pieces of hardware provide an idea that in the future all the sensor information will be unmanageable in a centralized infrastructure. Moreover, in the domain of social networks and online interactions, there is a continuous stream of events that is used for computing trending topics or word counting. Such an analysis could also be applied to all interactions occurring on the Internet. Clickstream analysis and software logs are two examples of Internet-scale data generated continuously. Considering that massive data processing can be spread across geographically distributed machines, processing could be aggregated using a global large-scale infrastructure in real-time [2].

Stream Processing Engines (SPEs) are designed to deal with real-time processing of high volume data streams. SPEs have evolved from centralized solutions [3], to be able to distribute queries among several nodes [4][5][6], to finally distribute operators (or processing elements) that solve a query across different nodes [7][8]. The latter type is especially interesting since there are cases where a single machine cannot cope with the processing of one operator.

SPEs use a graph-oriented paradigm, where vertices represent operators, also called Processing Elements (PEs), and the edges represent flows of data. An application defines the PEs and their interaction through a graph. The PEs can filter, map, unite, aggregate data, or carry out more complex processing. SPEs can cope with thousands of events per second, however, processing large geographically distributed data requires movement of data across the network, increasing the traffic and compromising the real-time results.

Processing large scale streams requires close to real-time global responses and a highly scalable infrastructure. The volume of data changes over time and the lost of a small amount of data is not critical to the results. However, SPEs do not provide tools that facilitate sharing and reuse of processed stream between clusters that perform the same task. In this work, we propose a model to share streams of geographically distributed data in a scalable manner.

The contribution of this work is a model that organizes SPEs into a Distributed Hash Table (DHT) structure in order to maintain scalable localization of resources. The system uses a Publish/Subscribe system to find and share streams and avoid reprocessing the events. This is a scalable, fault-tolerant and self-organized infrastructure, which maintains low latency using locality aware techniques. The model enables users to estimate latency before deciding whether to use the processed stream found in the system. This is a Quality of Service (QoS) measure, in order to cope with real-time restrictions.

The remainder of this article is organized as follows: Section II presents our system model giving details about each component and Section III details the processing steps. We discuss related work in Section IV and finally, present concluding remarks in Section V.

II. System Model

In traditional stream processing systems, each application is independent and works isolated from other applications producing data re-processing, which wastes resources. We propose a system model to process massive data streams in a distributed and collaborative manner. Our goal is to provide an infrastructure capable of dealing with the overwhelming amount of data available from diverse sources. Participants may share the pre-processed data streams, in order to avoid reprocessing of same data by other participants. We claim that this solution can be helpful for building complex applications, which exploits the output data streams of smaller applications. Small applications could provide their results as an input to more complex applications, enabling a more efficient use of processing. This scenario is presented in Figure 1. In the figure, application 2 (App 2) is a complex application, which can be built on top of the data pre-processed and shared by application 1 (App 1) and 3 (App 3).

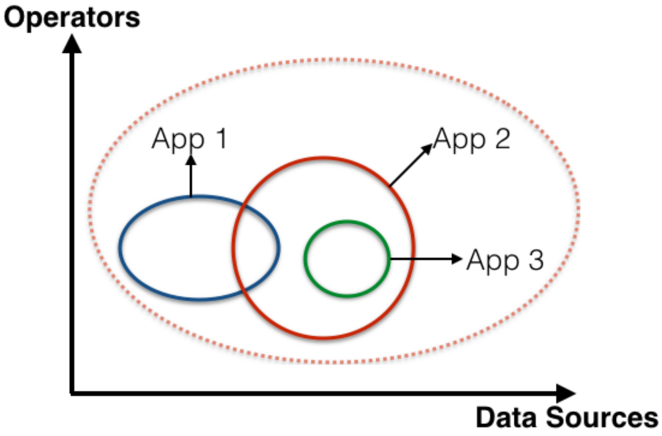


Figure 1. Stream processing and pre-processed data sharing

The most important challenges of implementing a large-scale SPEs infrastructure are: (1) scalability, the system must be able to process a large amount of data over several geographically distributed SPEs; (2) low latency, due to the real-time nature of SPEs, latency must be minimal despite the SPEs location; (3) fault tolerance, the system must be capable of dealing with failures and changing conditions of the communication network (latency, partitions, etc.).

We consider a scenario where multiple SPEs, geographically distributed around the world, collaborate by publishing their processed streams within the community. In our system, SPEs are organized into a DHT structure in order to maintain a scalable localization of resources. The system uses the Publish/Subscribe paradigm to publish and share data streams with remote SPEs. Data streams are identified by a description file, which provides detailed i stream treatment nformation. Publish/Subscribe has become a popular communication paradigm that provides a loosely coupled form of interaction among many publishing data sources and many subscribing data sinks. In Publish/Subscribe paradigm, messages are published into channels or topics asynchronously, without knowing the subscribers. On the other hand, the subscribers state their interest in one or more topics, and receive messages without knowing the publishers. This decoupling of publishers and subscribers enables greater scalability and a dynamic network topology.

A. Layered view

We propose an architecture composed of SPEs organized over a DHT-based P2P network where peers or SPEs share their resources in order to reduce data re-processing. From now onwards, we consider a peer as an instance of a SPE. The proposed architecture is composed of 4 layers: the overlay network, a stream sharing system, the stream processing engine and a latency aware tool. The first component provides efficient data localization. The second is responsible of publishing the available pre-processed streams. The third is able to process streams, and finally, the fourth component is in charge of estimating the latency of the data movement when sharing the streams. Figure 2 presents the proposed architecture, which is detailed below.

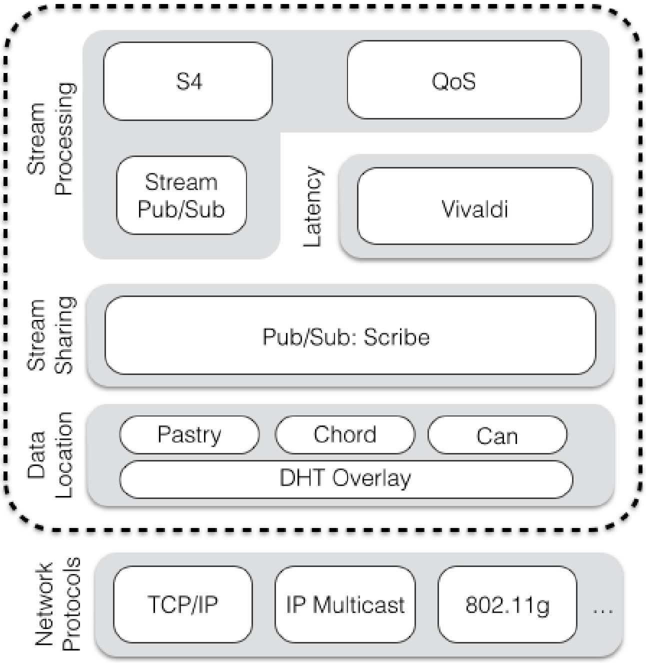


Figure 2. Distributed SPE (DSPE) architecture

B. Overlay Network

The DHT network is implemented using Pastry [9]; however, any other DHT, such as Chord [10], can be used in its place. Pastry is a well-known KBR (key-based routing), which provides scalable and efficient data localization. Pastry routing can efficiently locate data in a logarithmic number of routing hops $logN$, where N is the number of peers in the network. DHT-based overlays like Pastry can manage millions of participants without compromising performance, providing the substrate to build large scale systems.

Every peer in Pastry [9] is assigned a unique node ID in a space of 128-bit identifiers generated using a cryptographic hash SHA-1. The neighbors of a peer in Pastry are stored in the *leafset* that contains the L numerically closest peers, $L/2$ clockwise and $L/2$ counterclockwise. Pastry routing algorithm is a prefix-based algorithm that routes a message to the numerically closest peer of a given key k, we call this peer the *responsible* for k. The Pastry routing table stores on the n^{th} row the IP address of peers whose nodeIDs share the first n digits with the nodeID of the present peer. The algorithm forwards the messages to a peer chosen from its routing table that shares at least one more digit with the key k than the current peer. If no such peer can be found and the current peer does not know any other peer that is numerically closer to k, then the current peer is responsible for k and the routing ends.

Pastry has the advantage over other DHTs of including a neighbor list, which maintains contacts to peers close in terms of a metric, for example latency. In our case this improves routing and performance.

C. Sharing Streams

Stream sharing is achieved by exploiting a Publish/Subscribe mechanism specially suited for DHT networks, called Scribe. Scribe [11] is a topic-based system built on top of Pastry [9] that creates a multicast tree, which contains all the peers subscribed to a given topic. The multicast tree is essential to notify subscribers about updates on the given topic.

Each topic is referenced to by an identifier and the Pastry node with the closest identifier to the topic becomes its responsible peer. A multicast tree is built for each topic, rooted at the corresponding responsible peer. In Scribe when a new node subscribes to a subject, its subscription is routed by Pastry to the corresponding responsible peer. The nodes in the path towards the responsible peer update the tree structure in order to include the new subscriber in a distributed manner. When an event is published for a subject or topic, a message is routed through Pastry to the peer responsible for that subject. The responsible peer is addressed by the subject's identifier.

DSPEs can publish their streams identifying them using the stream data source. Then any operator or subset of operators related to that data source will be published at the same peer. Subscribers can join the group by performing the subscribe operation using the data source of their interest.

When an event arrives at the responsible peer for a given topic, a matching process among the description files of the streams is performed in order to find streams that match the query. Then, the references to the candidates' streams are returned.

D. Stream Processing

Stream processing has generated the attention of scientific community in the last years, arising as a promissory solution to process the huge amount of data generated nowadays. Many SPEs have been developed, systems like S4 [7], Storm of Twitter [8], TimeStream [12], StreamCloud [13], SEEP [14], D-Stream [15], MillWheel [16], Kinesis [17] among others, are systems proposed to process massive data in real-time. In this work, we focus on the Apache solution, called Simple Scalable Streaming System (S4).

S4 [7] is a general-purpose, distributed, scalable, event-driven, modular platform that allows programmers to easily implement applications for real-time processing of continuous unbounded streams of data. SPEs like S4 have a graph-oriented programming model where nodes represent operators, also called processing elements (PE), and the edges represent data flows. A query defines the PEs and their interaction. PEs can filter, map, unite, aggregate data, or carry out more complex processing. PEs are the basic computational unit in S4. They consume events on the basis of keys and may generate results as events. PEs are executed on Processing Nodes (PN), which are machines in a cluster. A special type of PEs called *adapters* associate tuples with keys.

S4 can be deployed on commodity hardware achieving low latencies in communication. S4 uses a push model where events are pushed to the next PE as fast as possible. In case a PE becomes overloaded, S4 uses load shedding and, in case of failure, S4 provides state recovery via uncoordinated checkpointing, using a coordinated communication system to detect node failures and notify nodes.

E. Low Latency

In P2P systems, participants are distributed all over the world, experiencing different communication latencies. Neighbors on a DHT can have greater communication latency compared to farther located peers on the DHT.

Due to the online nature of stream processing it is essential to reach low latency responses. For this reason, is important to provide information about pre-processed data latency in order to make the decision of exploiting such information or reprocessing it locally.

Applications have to meet different QoS requirements, furthermore the access to remote pre-processed streams experience different latencies. Applications must be able to evaluate the performance of using pre-processed data as an input stream. To cope with this requirement, our model provides a QoS module that estimates latency based on the Vivaldi algorithm [18].

Vivaldi [18] is an algorithm to estimate distance between peers in a fully distributed manner. It is based on the principle of spring relaxation to find minimal energy configurations in the system measuring latencies. Vivaldi presents a fully distributed lightweight algorithm that assigns synthetic coordinates to nodes in such a way that the distance between the coordinates of two nodes accurately predicts the communication latency between the nodes.

Vivaldi does not require a fixed network infrastructure or especial nodes to compute distances. Instead, any node can compute good quality coordinates by collecting latency information from only a reduced number of nodes. To collect information, Vivaldi piggybacks data on communication messages enabling traffic reduction while keeping other nodes informed on latencies experienced. The use of communication messages to spread information enables Vivaldi to scale to a large number of nodes.

Vivaldi can be applied on P2P systems in a straightforward manner. Dabek et al. have applied Vivaldi over a Chord [10] infrastructure to reach a low latency service over a P2P network [19]. Steiner and Bliersack have analyzed Vivaldi's performance [20], reinforcing Vivaldi authors' claims about the accuracy and the ability to scale of the algorithm. However, they also conclude that Vivaldi is not suitable for selecting close-by peers (within the same ISP). Round-trip time is composed of three elements: propagation delay, transmission delay, and queuing delay. On close-by peers RTT is small and become masked by the other components inducing noise to the estimation process degrading the estimation accuracy. This is not the case of our work since our scenario considers worldwide distributed SPEs.

QoS module estimates latency for the candidate streams provided by Scribe. Once latency is estimated, the stream processing layer can decide whether to exploit the remote pre-processed data or to start the reprocessing of data locally.

III. Processing Model

Given a DSPE requiring processing a distributed data stream, the processing model follows 4 steps:

1) **Stream Publication**
DSPEs process data streams generating an output result. This output is a stream that can be shared as pre-processed data to be exploited by other applications. The sharing process relies on publishing data about the output streams in order to allow another SPE to know if this is the data it needs.
The data about the stream to be shared is:
   - *Data source*
   - *Description of the processing*
   - *IP address of the SPE.*

   The identifier of the stream is defined by the *data source* or input stream the application receives. The Publish/Subscribe mechanism sends the stream data to a responsible peer which stores all the streams related to that same identifier or topic, using `SHA(identifier)`. Additional information or metadata could be published in order to facilitate the matching process and also to determine QoS characteristics based on the DSPE localization or bandwidth.
   Figure 3 presents an example of the publication of a pre-processed stream which identifier is the string `twitter` (data source). Scribe computes the `SHA(twitter)` and routes the data of the stream, description and IP address to the peer closest to the result of that computation. The peer uses `Scribe.publish(identifier, IP, description)` in order to publish the stream.
   The responsible peer R multicasts a message with this data to all the subscribers of the data source, to new subscribers and in case of updates.
2) **Resource Discovery**
DSPEs can locate SPEs that process the data of a specific stream subscribing the topic built for the data source. When the peer is subscribed to the data source it can receive new data about SPEs that are working on this data. `Scribe.join(identifier)` subscribes the peer to the correspondent multicast tree.
Each time a new stream is published, subscribed SPEs are notified of the updates in the topic. The node can decide locally if it is interested in one SPE output. Figure 3 shows peer P joining the group of the data source, called `twitter`, using Scribe.
3) **Data Sharing and Processing**
Data processing involves processing data either locally or exploiting pre-processed data from a remote DSPE.
A SPE that process a stream that is required by several others, builds a Scribe multicast tree with the subscribers that need this output stream (`Scribe.create(IP,data_source)`). This same node is the root of the multicast tree and the requesters use the IP address of the peer in order to subscribe to the stream. In this way, the source node does not send the stream directly to all the subscribers, but it uses the multicast tree to balance the load.
Figure 3 shows this step where the peer P joins the multicast tree of the stream generated by DSPE.
4) **Latency Estimation**
Once one stream is selected, the peer estimates the latency that the peer will experience during the retrieval of the stream from that remote DSPE. Latency is critical for online processing, however applications have different QoS requirements which should be considered. If it does not achieve the expected latency compared to the direct use of the data source, then the remote pre-processed data is discarded and the peer P should leave the DSPE group.

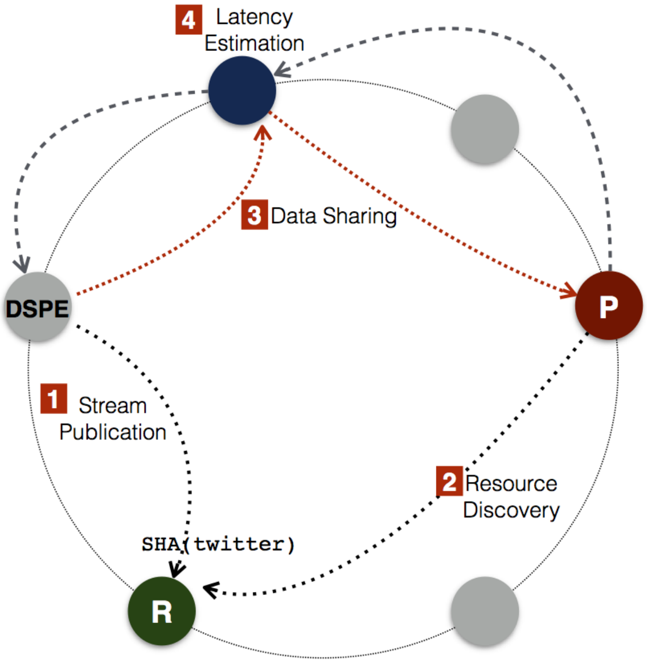


Figure 3. Distributed sharing-based model for stream processing

In Figure 3, the distributed stream sharing process is presented. The first step consists in checking the published streams. Secondly, the matching process between the query and the available streams are performed in order to select the candidate streams. Third, candidate streams are returned to the querying DSPE. Then, latency estimation takes place in order to discard or take preprocessed streams. Finally, the selected candidate stream is retrieved from the remote DSPE.

IV. Related Work

Recently, several platforms have built on top of SPEs to provide more functionality, differing from sharing stream of data among clusters of nodes. One of them is Trident [21], which is a high level abstraction on top of Storm that simplifies the process of building topologies using a micro-batching processing model. Spark Streaming [22] is a framework that similarly to Trident, that uses microbatch processing. Spark receives data from different sources and includes stateful operators to the SPE. Kafka [23] is a publish/subscribe system that provides log functionality to SPEs, which is designed for real-time activity. The tuple Kafka-Storm or Kafka-Spark has been proposed in order to guarantee fault tolerance. Below, we discuss related work comparing their main characteristics with our model.

Synergy [24] is a middleware for distributed stream processing systems that uses an overlay mesh network for communication. The distributed processing systems can use the whole architecture and different processing elements can be found at different nodes. They proposed the use of a DHT structure to store and share stream, for this goal we use a Publish/Subscribe system, which allows easy localization of stream about the same topic. For QoS, Synergy uses a process called *impact projection* in order to find a candidate set for processing. We build our system using Vivaldi [18] in order to maintain locality-aware stream processing.

SBON [25] is a layer between a stream processing system and the physical network that manages operator placement for stream processing. SBON uses space coordinate distance between two nodes to represent the overhead of query placement and the cost of routing data between them. In this case, the participants in the distributed systems can place operators of its own application to other nodes in the system. This is a different scenario than the one targeted in our work. We consider that different applications are communicated through a DHT and share their streams through a Publish/Subscribe system. Applications may belong to different owners and the sharing process does not use more resources in processing.

SensWeb [26] is an infrastructure for geocentric exploration of sensor data stream, which allows sharing data streams across multiple applications. We follow this same goal, however we aim at sharing processed streams produced as output of SPEs. SensorWeb is focused on map visualization, which is achieved through a coordinator and an indexing engine. Our work is focused on distributed scalable infrastructure achieved with a DHT and Publish/Subscribe middleware.

GATES [27] is a system that uses a grid middleware for processing distributed data stream. GATES system uses Open Grid Services Architecture (OGSA) to provide self-resource discovering. Our work does not maintain grid boundaries, and follows a P2P architecture to achieve the same.

In the grid category, we also found StreamGlobe [28], a system that classifies peer as super-peers and thin-peers to be able to manage and optimize large networks. Users register subscriptions and data stream at these interfaces. The StreamGlobe scheme uses a hierarchical architecture and uses the same framework as GATES to achieve resource discovery.

Branson et al. propose CLASP [29], a middleware that enables autonomous stream analysis systems to interoperate, providing them with opportunities for data access. In CLASP, applications that seek to cooperate, build virtual organizations that formalize permissible interoperation, called common interest policies (CIP). CIP specifies resources to share and each virtual organization defines a manager, a planner and coordinator process that support collaboration functions. CLASP can complement our work, since it defines a collaboration protocol and a system association. However, CLASP does not cover the distributed infrastructure, or the efficient resource discovery.

V. Concluding Remarks

In this paper, we identified the need of a distributed infrastructure to cope with the huge amount of data stream generated by diverse streaming data sources. We propose a distributed architecture, able to manage the data stream processing in a scalable way. Our architecture relies on a DHT network in which the SPEs communicate and coordinate their actions in order to cooperate to process data. Cooperation is done by sharing pre-processed data streams based on a Publish/Subscribe mechanism.

It is well known that DHT infrastructures have to deal with changing network conditions, affecting their communication latency. We tackle this problem by providing information about access latency to the pre-processed data resources. Such information allows the remote processing engine to decide whether to exploit the remote data stream or reprocess it locally.

We have generated a prototype of the architecture proposed in this article in the context of a Fondef IDEA grant, project code CA12i10314. Our future work is mainly focused on applying this architecture on data stream processing in the context of disaster scenarios.

Acknowledgment

The authors would like to thank the University of Santiago research project PMI-USA 1204 and STIC-AmSud project RESPOND 13-STIC 11. Erika Rosas thanks to CONICYT PAI/ACADEMIA 791220011.

References

[1] “Libelium world.” [Retrieved: May, 2015] http://www.libelium.com/smart_cities/

[2] S. Madden and M. van Steen, “Guest editors’ introduction: Internet-scale data management,” IEEE Internet Computing, vol. 16, no. 1, 2012, pp. 10–12.

[3] D. J. Abadi et al., “Aurora: a new model and architecture for data stream management,” VLDB J., vol. 12, no. 2, 2003, pp. 120–139.

[4] D. J. Abadi et al., “The design of the borealis stream processing engine,” in Second Biennial Conference on Innovative Data Systems Research (CIDR 2005), Asilomar, CA, January 2005, pp. 277–289.

[5] S. Krishnamurthy et al., “Telegraphcq: An architectural status report,” IEEE Data Eng. Bull., vol. 26, no. 1, 2003, pp. 11–18.

[6] A. Arasu et al., “STREAM: the stanford stream data manager,” IEEE Data Eng. Bull., vol. 26, no. 1, 2003, pp. 19–26.

[7] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, “S4: Distributed stream computing platform,” in Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, ser. ICDMW ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 170–177.

[8] “Storm.” [Retrieved: May, 2015] https://github.com/nathanmarz/storm/wiki

[9] A. I. T. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” in Middleware, ser. Lecture Notes in Computer Science, R. Guerraoui, Ed., vol. 2218. Springer, 2001, pp. 329–350.

[10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, ser. SIGCOMM ’01. New York, NY, USA: ACM, 2001, pp. 149–160.

[11] M. Castro, P. Druschel, A. M. Kermarrec, and A. I. Rowstron, “Scribe: A large-scale and decentralized application-level multicast infrastructure,” IEEE J.Sel. A. Commun., vol. 20, no. 8, Sep. 2006, pp. 1489–1499.

[12] Z. Qian et al., “Timestream: reliable stream computation in the cloud,” in Proceedings of the 8th ACM European Conference on Computer Systems, ser. EuroSys ’13. New York, NY, USA: ACM, 2013, pp. 1–14.

[13] V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, and P. Valduriez, “Streamcloud: An elastic and scalable data streaming system,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 12, 2012, pp. 2351–2365.

[14] R. Castro Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch, “Integrating scale out and fault tolerance in stream processing using operator state management,” in Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’13. New York, NY, USA: ACM, 2013, pp. 725–736.

[15] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, “Discretized streams: fault-tolerant streaming computation at scale,” in SOSP, M. Kaminsky and M. Dahlin, Eds. ACM, 2013, pp. 423–438.

[16] T. Akidau et al., “Millwheel: Fault-tolerant stream processing at internet scale,” in Very Large Data Bases, 2013, pp. 734–746.

[17] “Amazon kinesis.” [Retrieved: May, 2015] http://aws.amazon.com/kinesis/

[18] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, “Vivaldi: a decentralized network coordinate system,” SIGCOMM Comput. Commun. Rev., vol. 34, no. 4, Aug. 2004, pp. 15–26.

[19] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris, “Designing a dht for low latency and high throughput,” in Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1, ser. NSDI’04. Berkeley, CA, USA: USENIX Association, 2004, pp. 85–98.

[20] M. Steiner and E. Biersack, “Where is my peer? evaluation of the vivaldi network coordinate system in azureus,” in NETWORKING 2009, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5550, pp. 145–156.

[21] “Trident.” [Retrieved: May, 2015] https://storm.apache.org/documentation/Trident-API-Overview.html

[22] “Spark streaming.” [Retrieved: May, 2015] https://spark.apache.org/streaming/

[23] “Kafka.” [Retrieved: May, 2015] http://kafka.apache.org/

[24] T. Repantis, X. Gu, and V. Kalogeraki, “Synergy: Sharing-aware component composition for distributed stream processing systems,” in Middleware, ser. Lecture Notes in Computer Science, M. van Steen and M. Henning, Eds., vol. 4290. Springer, 2006, pp. 322–341.

[25] P. R. Pietzuch, J. Ledlie, M. Mitzenmacher, and M. I. Seltzer, “Network-aware overlays with network coordinates,” in 26th International Conference on Distributed Computing Systems Workshops (ICDCS 2006 Workshops), 4-7 July 2006, Lisboa, Portugal. IEEE Computer Society, 2006, p. 12.

[26] L. Luo, A. Kansal, S. Nath, and F. Zhao, “Sharing and exploring sensor streams over geocentric interfaces,” in 16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2008, November 5-7, 2008, Irvine, California, USA, Proceedings, 2008, p. 3.

[27] L. Chen, K. Reddy, and G. Agrawal, “Gates: A grid-based middleware for processing distributed data streams,” in HPDC. IEEE Computer Society, 2004, pp. 192–201.

[28] R. Kuntschke, B. Stegmaier, A. Kemper, and A. Reiser, “Streamglobe: Processing and sharing data streams in grid-based p2p infrastructures,” in VLDB, K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P.-Å. Larson, and B. C. Ooi, Eds. ACM, 2005, pp. 1259–1262.

[29] M. Branson, F. Douglis, B. Fawcett, Z. Liu, A. Riabov, and F. Ye, “Clasp: Collaborating, autonomous stream processing systems,” in Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware, ser. Middleware ’07. New York, NY, USA: Springer-Verlag New York, Inc., 2007, pp. 348–367.

Autonomic Metaheuristic Optimization
with Application to Run-Time Software Adaptation

John M. Ewing and Daniel A. Menascé

Department of Computer Science, MS 4A5
The Volgenau School of Engineering, George Mason University
Fairfax, Virginia, United States of America
Email: jewing2@gmu.edu,menasce@cs.gmu.edu

Abstract—**This paper presents a general meta-optimization approach for improving self-optimization in autonomic systems. This approach can improve optimization performance and lower costs by reducing human effort needed to tune optimization algorithms. We apply our meta-optimization approach to Self-Architecting Software Systems (SASSY). A genetic algorithm is used to meta-optimize both the architecture search module and the service selection search module in SASSY. Four different heuristic search algorithms (hill-climbing, beam search, evolutionary programming, and simulated annealing) are made available to be meta-optimized in both the architecture search module and the service selection search module. This meta-optimization process generated twelve new heuristic search algorithm pairs for solving SASSY optimization problems. In a large set of simulation experiments, two of the generated heuristic search algorithm pairs provided superior performance to the control (which was the previously best heuristic search algorithm pair known in SASSY).**

Keywords–Intelligent systems; Autonomous agents; Evolutionary computation; Genetic algorithms

I. Introduction

Autonomic computing is a discipline that studies the design of methods and techniques that enable information systems to manage themselves. The self-management capabilities can be broken down into four self-* properties: self-configuration, self-optimization, self-healing, and self-protection [1]. A driving force in the adaptation of autonomic computing is the desire to reduce the Total Cost of Ownership (TCO); autonomic computing achieves this goal by reducing maintenance costs, in particular the level of effort required by system administrators.

Achieving each of the self-* properties presents special challenges. In this work, we focus on the challenges presented by run-time self-optimization in the face of changes in the environment. Autonomic systems that perform self-optimization require some computational method to discover a configuration or a sequence of actions that will optimize the system. A number of techniques including linear programming, heuristic search, and machine learning have been employed to conduct self-optimization in autonomic systems [2][3][4]. Most self-optimizing autonomic systems share the following three considerations:

1) multiple optimization problems will be encountered over the life of the autonomic system,
2) encountered optimization problems must be solved in near real-time, and
3) the performance of the optimization algorithm is impacted by parameters that control the behavior of the algorithm.

For many autonomic systems, it is reasonable to expect that hundreds to thousands of optimization problems will be encountered over the system's lifetime. Self-optimization is often invoked in support of self-healing; restoring functionality to a system requires expeditious decision-making on the part of the optimizing algorithm.

Optimization conducted through heuristic search algorithms can have widely varying performance. The performance of a heuristic search algorithm largely depends upon the type of algorithm and its attendant parameter settings. The topology of the system's objective function over the system's configuration space interacts heavily with the selection of the heuristic search algorithm and attendant parameters. These interactions can be difficult to predict, and require human system administrators with significant knowledge, experience, and time to set them correctly. This additional effort can substantially reduce the original cost savings provided by the autonomic system.

To reduce costs and improve the performance of self-optimizing systems, we propose a meta-optimization technique for autonomic systems. Meta-optimization is particularly well-suited to self-optimizing autonomic systems for two reasons:

- A meta-optimized optimization algorithm is likely to yield improved results each time the algorithm is invoked. The cumulative positive impact of making better decisions over the system's lifetime can be substantial.
- Optimizations can be solved in a matter of seconds, therefore it is computationally feasible to execute the optimization algorithm thousands of times either offline or between self-optimization events.

Huebscher and McCann [5] propose classifying systems based on their degree of autonomicity. The authors suggest five levels of autonomicity:

1) *Support*–At this lowest level of autonomicity, a system focuses on only a subset self-* properties and/or focuses only on a subset of components.
2) *Core*–A system with core autonomicity enables self-* properties on all components but provides no method for modifying system goals online.
3) *Autonomous*–An autonomous system enables self-* properties on all components but does not possess awareness of the autonomic manager's performance.

4) *Autonomic*–An autonomic system enables self-* properties on all components, is aware of the autonomic manager's performance, and can adapt the behavior of the autonomic manager to improve performance.
5) *Closed-Loop*–A system with closed-loop autonomicity enables self-* properties on all components, is aware of the autonomic manager's performance, and grows the capabilities of the autonomic manager through intelligent reasoning.

Applying meta-optimization can contribute to the transformation of autonomous systems into autonomic systems.

This paper makes the following three contributions:

1) a framework for conducting meta-optimization on self-optimizing systems,
2) a demonstration of the framework on an application using SASSY, and
3) an experimental evaluation of the resulting meta-optimized heuristic search algorithms.

The organization of this paper is as follows. Section II provides a brief overview of the SASSY project that motivated the need for the development of the ideas presented in this paper. Section III formalizes the meta-optimization problem. The following section presents the meta-optimization framework. Section V presents and discusses the results of our experimental evaluation. The following section discusses related work and Section VII presents some concluding remarks.

II. Overview of SASSY

In previous work, we presented an autonomic framework for managing Service Oriented Architecture (SOA) applications called SASSY [6][7]. SASSY optimizes the performance of systems by modifying architectural patterns and changing service provider (SP) selections.

In SASSY, a user defines data flows among activities for a new SOA application via a graphical interface [6]. The user can specify multiple Quality of Service (QoS) requirements associated with the framework. These QoS requirements are termed service sequence scenarios (SSS) and they couple a desired QoS goal with a path through the data flows. The degree of satisfaction of the QoS goals is reflected in a global utility function, U_g, which serves as the objective function in SASSY's self-optimization processes. A detailed description of how data flows and SSSs are defined in the SASSY framework can be found in [3] and [6]. It is worth noting that the global utility functions are typically concave with multiple optima.

SASSY generates a base software architecture from the user's requirements that consists of a coordinator and a basic software component for each activity defined in the data flow. The coordinator is linked to each basic software component and SSS performance models are automatically produced using expression trees and the set of rules described in [6].

This base architecture can be modified through the substitution of a basic component with a composite component. A composite component uses multiple SPs and is created from an architectural pattern template. For example, a composite component might be constructed from a load balancing architectural pattern template; the composite component might use two different SPs and distribute the offered load according to the SPs' advertised capacities [8].

To make the architecture executable, the coordinator must bind a set of SPs to the basic components in the architecture. Different SPs may offer the same service with varying levels of performance and cost. For a given architecture, SASSY searches for a combination of SPs that maximizes U_g.

The coordinator is able to substitute patterns and components to the architecture at run-time [9]. This enables the system to re-architect at run-time when new services become available or a service currently bound to the architecture fails.

TABLE I. SSSes USED IN EXPERIMENTAL EVALUATION.

QoS Metric	Weight	Number of Activities
Security Option 1	0.08	16
Security Option 1	0.03	9
Security Option 2	0.11	11
Security Option 2	0.07	9
Throughput	0.11	11
Throughput	0.06	16
Throughput	0.02	11
Availability	0.12	16
Availability	0.08	11
Availability	0.04	16
Availability	0.04	11
Execution Time	0.18	11
Execution Time	0.03	16
Execution Time	0.03	11

Our previous work considers small- to medium-sized data flows in SASSY with up to 30 activities [3][6]. Here, we consider the much larger SOA application shown in Figure 1 that has 65 activities. A summary of the SSSes defined for this application can be found in Table I. For each SSS, the table shows its QoS metric, the weight of that metric in the computation of the global utility U_g, and the number of software components of that SSS. The heuristic search optimization algorithms considered in our previous work were tuned on an application with 30 activities. In this paper we apply a meta-optimization process to determine if more suitable heuristic search algorithms can be found for this larger application.

III. Examining Meta-Optimization

All self-optimizing systems have methods for judging the efficacy of a given configuration or sequence of actions. For the purposes of expediency in discussion, we assume that all self-optimizing systems can be gauged with a global utility function.

Formally, self-optimization can be specified as:

Find a system state S^* such that

$$S^* = \text{argmax}_S \; U_g(S, \mathcal{K}). \tag{1}$$

where $U_g()$ is a global utility function representing the usefulness of being at system state S when the operating environment is at state $\mathcal{K}$.

To achieve optimization, self-optimizing autonomic systems either employ approximate optimization algorithms or make restrictions in the number of system states that may be considered. Equation (2) shows the optimization process, B, producing an approximately optimized state, S_a^* with optimization algorithm, $\mathcal{H}$.

$$S_a^* = B(\mathcal{H}, \mathcal{K}). \tag{2}$$

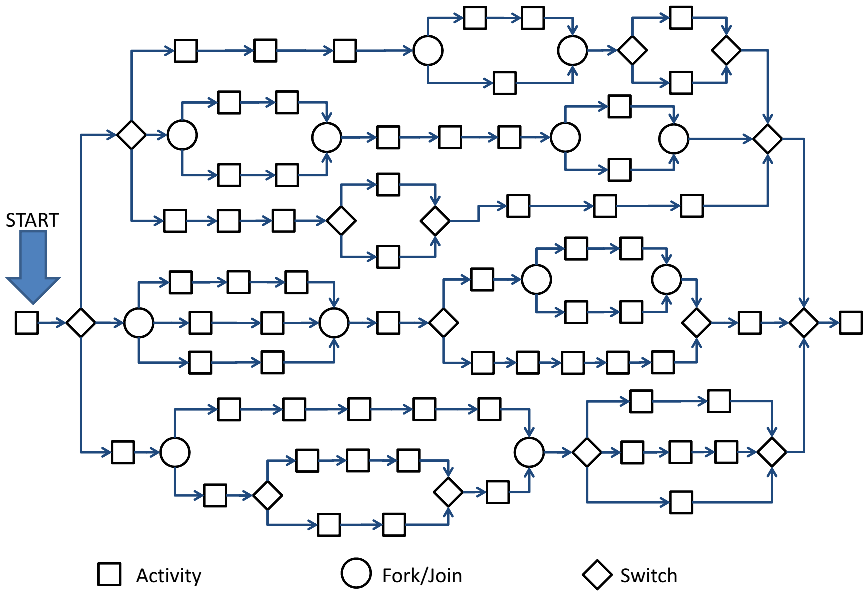


Figure 1. SOA application with 65 activities.

Often, these approximate optimization algorithms are nondeterministic due to stochastic operations (e.g., mutations in evolutionary algorithms). Thus, to measure the performance of an optimization algorithm $\mathcal{H}$, its expected global utility $\overline{U}_{\mathcal{H}}$ over multiple executions of B should be considered:

$$\overline{U}_{\mathcal{H}} = \mathbb{E}\ [U_g(S_a^*)] = \mathbb{E}\ [U_g(B(\mathcal{H}, \mathcal{K}))]. \tag{3}$$

The meta-optimization problem can be formally specified as follows:

Find an approximate optimization algorithm $\mathcal{H}^*$ such that

$$\mathcal{H}^* = \text{argmax}_{\mathcal{H}}\ \mathbb{E}\ [U_g(B(\mathcal{H}, \mathcal{K}))] \tag{4}$$

$$t_{\mathcal{H}} \leq t_L \tag{5}$$

where $t_{\mathcal{H}}$ is the execution time for $\mathcal{H}$ and t_L is a time limit.

A. Meta-Optimization in SASSY

There are two NP-hard optimization problems that need to be solved in near real-time for SASSY [6]:

1) an architecture optimization problem and
2) a service selection optimization problem.

The two optimization problems are in fact nested: before an individual architecture can be evaluated, an approximately optimal service selection must first be found.

Formally, the SASSY optimization problem can be expressed as:

Find an architecture $\mathcal{A}^*$ and a corresponding SP allocation Z^* such that

$$(\mathcal{A}^*, Z^*) = \text{argmax}_{(\mathcal{A}, Z)}\ U_g(\mathcal{A}, Z, \mathcal{K}). \tag{6}$$

where $U_g(\mathcal{A}, Z)$ is the global utility of architecture $\mathcal{A}$ and service selection Z, with the state of all SPs in the environment denoted by $\mathcal{K}$. This optimization problem may be modified by adding a cost constraint. In the cost-constrained case, there is a cost associated with each SP for providing a certain QoS level [6].

The optimization process, B, used by SASSY's centralized autonomic controller requires two algorithms: $\mathcal{H}_{\mathcal{A}}$ for the architecture search and $\mathcal{H}_Z$ for the service selection search. Equation (7) shows that the optimization process requires one more input, $\mathcal{A}_c$, the current architecture. This provides a useful starting position for the algorithm $\mathcal{H}_{\mathcal{A}}$, since the $\mathcal{A}_c$ is often close to an architecture $\mathcal{A}_a^*$.

$$(\mathcal{A}_a^*, Z_a^*) = B(\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z, \mathcal{A}_c, \mathcal{K}) \tag{7}$$

The performance of the algorithm pair, $\overline{U}_{\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z}$, is expressed below:

$$\begin{aligned} \overline{U}_{\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z} &= \mathbb{E}\ [U_g(\mathcal{A}_a^*, Z_a^*)] \\ &= \mathbb{E}\ [U_g(B(\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z, \mathcal{A}_c, \mathcal{K}))]. \end{aligned} \tag{8}$$

Equation (9) describes the meta-optimization problem in SASSY:

Find a pair of approximate optimization algorithms $(\mathcal{H}_{\mathcal{A}}^*, \mathcal{H}_Z^*)$ such that

$$\begin{aligned} (\mathcal{H}_{\mathcal{A}}^*, \mathcal{H}_Z^*) \;=\; \text{argmax}_{(\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z)}\ \mathbb{E}\ [U_g(B(\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z, \\ \mathcal{A}_c, \mathcal{K}))] \end{aligned} \tag{9}$$

$$t_{(\mathcal{H}_{\mathcal{A}}, \mathcal{H}_Z)} \leq t_L \tag{10}$$

SASSY can employ a number of heuristic search methods as approximate optimization algorithms in solving the architectural pattern problem and the SP selection problem. Hill-climbing, beam search, simulated annealing, and evolutionary programming have been implemented and tested in the SASSY autonomic controller with varying degrees of effectiveness [3]. Each of these heuristic search algorithms requires multiple parameter settings that can have potentially large impacts on the optimization process performance.

IV. Meta-Optimization Framework

Meta-Optimization in SASSY is currently an offline activity that requires some minimal supervision from a human administrator.

As demonstrated in (4) and (9), certain inputs are required in the meta-optimization process. In the general case, we require the operating environment state, $\mathcal{K}$, to conduct a meta-optimization. For SASSY meta-optimizations, we additionally require the system's current architecture, $\mathcal{A}_c$.

To ensure acquisition of appropriate meta-optimization inputs, we propose the following three-step meta-optimization process:

1) capture candidate sample problem set,
2) select finalist problems from candidate problem set, and
3) apply meta-optimization procedure to finalist problems.

A candidate sample problem set is a pool of observed or generated optimization problems. A candidate sample problem set may be large, and it may not be computationally feasible to conduct effective meta-optimization on each problem in this set. When the candidate problem set is large, a method is required for selecting a promising subset (i.e., the finalists) of the candidate problems. A meta-optimization procedure can then be pursued on the small set of finalist problems.

A. Generating Candidate Problems in SASSY

To capture a candidate sample problem set in SASSY, we execute the SASSY system in a simulated service environment. The simulation generates SP failures, SP degradations, and SP repairs. If an SP failure or SP degradation reduces U_g below some threshold, the autonomic controller will initiate an optimization process to find a new architecture, $\mathcal{A}$, and SP selection, Z. When the performance monitor detects SP repair events, the autonomic controller will also initiate an optimization process to determine if a better $\mathcal{A}$ and Z can be achieved. The candidate problem set is produced by collecting randomly sampled problems encountered in the simulation—the purpose is to avoid oversampling small portions of the problem space.

In the SASSY application depicted in Figure 1, we randomly generated between three and ten SPs for each of the 65 activities, yielding an overall total of 404 SPs. We conducted a relatively long initial optimization search to find a near-optimal starting architecture, $\mathcal{A}_i$, and a near-optimal SP selection, Z_i. We instantiated a SASSY system using the beam search/evolutionary programming `BS-EP` heuristic search algorithm pair from [3]. Starting the SASSY system with $\mathcal{A}_i$ and Z_i, we simulated SP failures, SP degradation, and SP repair events over time. We conducted 26 such simulations and captured 1% of the encountered optimization problems by the SASSY autonomic controller. This process generated 1,041 candidate sample problems.

B. Selecting Finalist Problems in SASSY

Our previous work [3] demonstrated that sometimes a small fraction of SASSY optimization problems are particularly challenging. The choice of heuristic search algorithms on these challenge problems can have an outsized impact on the SASSY system's overall performance. Identifying challenge problems with machine learning techniques has proven difficult [3]. To improve the odds of including one or more challenge problems in the finalist subset, we prioritize diversity when choosing finalists from candidates.

To develop a diverse finalist subset, we examine two summary statistics:

1) $\Delta\ U_g$ is the difference in U_g from the last optimization search. This measures the severity of the optimization problem.
2) $f_{\Delta}(\mathcal{K})$ is the fraction of SPs that have changed state due to failure, degradation, or repair since the last optimization search. This measures the degree of change in the environment.

Figure 2 shows a scatter plot of the 1,041 candidate problems using these summary statistics.

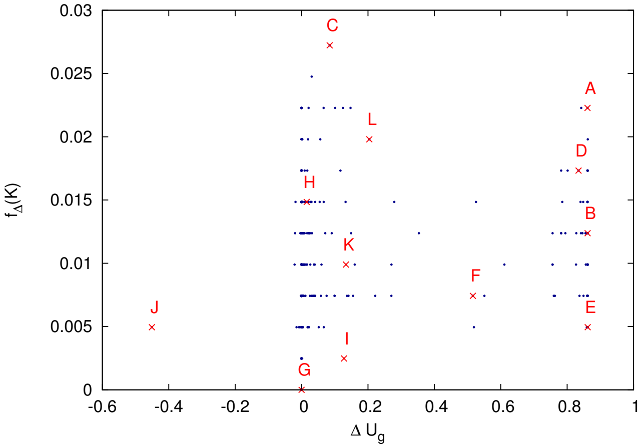


Figure 2. The candidate problem set plotted using summary statistics. The twelve finalist problems are labeled A-L and marked with red x's.

To pick a diverse group of finalist problems, we select problems distributed across the full range, including some outliers. Challenge problems may be uncommon, so it is not necessary that each finalist problem represent a cluster of candidate problems. The twelve finalist problems were selected by assessing Figure 2 and are labeled A through L.

C. Applying Meta-Optimization Procedure

Figure 3 describes the meta-optimization procedure we applied to the SASSY autonomic controller. Exactly one finalist sample problem is assigned to an instance of the meta-optimizer. The arrows departing from Box 1 show how the information captured in the finalist sample problem is distributed.

- The current architecture, $\mathcal{A}_c$, is sent to the Meta-Optimizer.
- The performance of the SPs in the environment is sent to the SSS Performance Modeler.
- A list of the available SPs in the environment is provided to the Service Selection Search Module.

The Meta-Optimizer (Box 2) generates a pair of heuristic search algorithms that are then provided to the Architecture Search Module (Box 3) and the Service Selection Search Module (Box 4). Additionally, the Meta-Optimizer directs the Architecture Search Module to commence an optimization

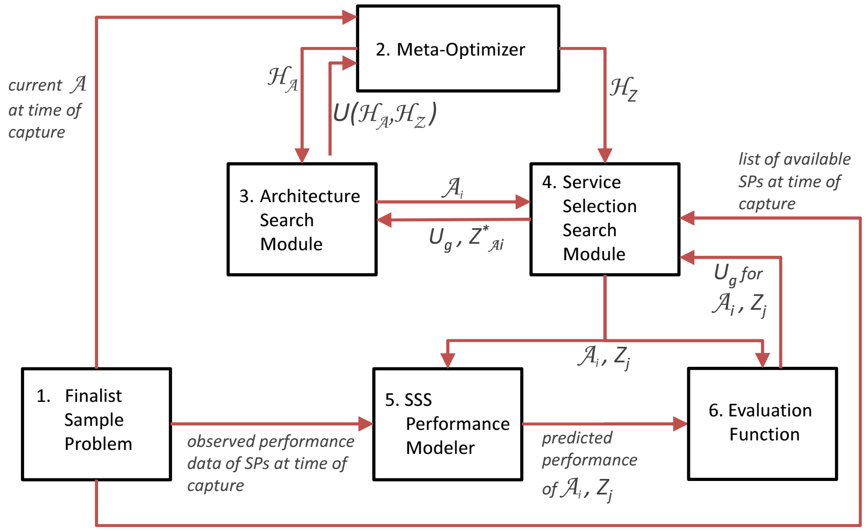

2. Meta-Optimizer
current $\mathcal{A}$ at time of capture
$\mathcal{H}_{\mathcal{A}}$
$U(\mathcal{H}_{\mathcal{A}},\mathcal{H}_Z)$
$\mathcal{H}_Z$
3. Architecture Search Module
$\mathcal{A}_i$
U_g, $Z^*_{\mathcal{A}i}$
4. Service Selection Search Module
list of available SPs at time of capture
U_g for $\mathcal{A}_i$, Z_j
$\mathcal{A}_i$, Z_j
1. Finalist Sample Problem
observed performance data of SPs at time of capture
5. SSS Performance Modeler
predicted performance of $\mathcal{A}_i$, Z_j
6. Evaluation Function

Figure 3. The meta-optimization procedure applied to SASSY.

search. The optimization search will be repeated n times before the Meta-Optimizer changes the heuristic search algorithms in the search modules (Boxes 3 and 4). The score for the heuristic search pair is the average predicted U_g of the returned $\mathcal{A}$ and Z.

The heart of the architecture/SP selection optimization is the interaction among boxes 3, 4, 5, and 6. When the architecture optimization search begins, the Architecture Search Module (Box 3) requests the Service Selection Search Module (Box 4) to find an optimal Z_i for a given $\mathcal{A}_i$. As it conducts the SP selection search, the Service Selection Search Module requests performance predictions for a given $\mathcal{A}_i$ and Z_j.

1) Genetic Algorithm for the Meta-Optimizer: We used a genetic algorithm as our meta-optimization algorithm for the following four reasons.

1) The genotype representation provides an elegant mechanism for representing complex objects.
2) The crossover and mutation operators can be applied to the genotype representation in a simple and uniform way.
3) Genetic algorithms are robust in the face of noisy evaluations.
4) The crossover operator can blend two different heuristic pair algorithms to explore the heuristic parameter space between them.

The heuristic search algorithms and their attendant parameters are encoded into binary strings. The format of these binary strings are defined in Table II and Table III. The genotype of the heuristic search algorithm pair is formed by concatenating these two binary strings. For a more detailed discussion of the heuristic search algorithm parameters, see [3].

The service selection search budget parameter, N_Z, in Table III refers to the number of SP selections to be evaluated for each architecture evaluation. Thus, the total number of model evaluations, $N_{\mathcal{M}}$ can be computed as follows:

$$N_{\mathcal{M}} = N_{\mathcal{A}} \times N_Z \qquad (11)$$

where $N_{\mathcal{A}}$ is the architecture search budget parameter.

In this work, the window for completing an architecture optimization search was set to be 7.5 seconds. On systems with two 2.4 GHz quad-core hyper-threading Intel Xeon processors this translated to $N_{\mathcal{M}} = 47,600$. Using this information, $N_{\mathcal{A}}$ was then derived from N_Z.

In most genetic algorithms, the size of the parent population and offspring population are equal (in this work we use a population size of 15). With each generation, the parent solutions are discarded, and the offspring become the next generation of parents.

Each new offspring is generated by probabilistically selecting two parents. We use the linear ranking method for parent selection [10][11]. In linear ranking, the population is first sorted in descending order according to fitness, $\overline{U}_{\mathcal{H}_{\mathcal{A}},\mathcal{H}_Z}$. The probability of selecting member i is:

$$P(i) = \frac{1+\mathcal{S}}{M} - \frac{2\mathcal{S}(i-1)}{M(M-1)} \qquad (12)$$

where $\mathcal{S}$ is a pressure selection variable that may take on values in the range of $[0,1]$. When $\mathcal{S}$ is zero, all members of the population have an equal chance of being selected; as $\mathcal{S}$ increases, the probability increases of selecting the fittest members of the population. Here, we use $\mathcal{S} = 1$, which should speed the final convergence on concave maxima—this is a desirable feature given limitations on time and resources for our meta-optimization.

The offspring is produced from the two parents through the uniform crossover operator with the crossover probability set to 0.08. The genetic algorithm transcribes the binary string from the first parent selected to the offspring. With each transcribed bit, there is an 8% chance that the genetic algorithm will swap the parents for the source of the transcription [10].

Once the crossover operation is complete for a new offspring, the bit-flip mutation operator is invoked. To avoid entrapment in hamming cliffs, the binary strings are converted into Gray code [12] before the bit-flip mutation operator is applied. The bit-flip mutation operator examines each bit of the genotype binary string and flips a given bit with a probability of 0.02.

TABLE II. COMPOSITION OF ARCHITECTURE SEARCH BINARY STRING.

Parameter	Algorithm	Type	Min	Max	Step	# bits
search algorithm	all	enum	N/A	N/A	N/A	2
hill-climbing mode	hill-climbing	enum	N/A	N/A	N/A	1
beam search mode	beam search	enum	N/A	N/A	N/A	2
neighborhood filtering	hill-climbing & beam search	boolean	N/A	N/A	N/A	1
# of SSSes in filter	hill-climbing & beam search	integer	1	13	1	4
# of components in filter	hill-climbing & beam search	integer	1	64	1	6
beam width	beam search	integer	2	5	1	2
parent population size	evolutionary programming	integer	1	20	1	5
brood size	evolutionary programming	floating point	1.0	8.5	0.5	4
overlapping population	evolutionary programming	boolean	N/A	N/A	N/A	1
initial step size	evolutionary programming	floating point	1.0	4.5	0.5	3
adaptive step factor	evolutionary programming	floating point	1.0	4.5	0.5	3
initial probability	simulated annealing	floating point	0.1	0.7	0.04	4
final probability	simulated annealing	floating point	0.00001	0.00016	0.00001	4

TABLE III. COMPOSITION OF SERVICE SELECTION SEARCH BINARY STRING.

Parameter	Algorithm	Type	Min	Max	Step	# bits
search budget, N_Z	all	integer	100	2500	25	7
search algorithm	all	enum	N/A	N/A	N/A	2
hill-climbing mode	hill-climbing	enum	N/A	N/A	N/A	1
beam search mode	beam search	enum	N/A	N/A	N/A	2
neighborhood filtering	hill-climbing & beam search	boolean	N/A	N/A	N/A	1
# of SSSes in filter	hill-climbing & beam search	integer	1	13	1	4
# of components in filter	hill-climbing & beam search	integer	1	64	1	6
beam width	beam search	integer	2	5	1	2
parent population size	evolutionary programming	integer	1	20	1	5
brood size	evolutionary programming	floating point	1.0	8.5	0.5	4
overlapping population	evolutionary programming	boolean	N/A	N/A	N/A	1
initial step size	evolutionary programming	floating point	1.0	4.5	0.5	3
adaptive step factor	evolutionary programming	floating point	1.0	4.5	0.5	3
initial probability	simulated annealing	floating point	0.1	0.7	0.04	4
final probability	simulated annealing	floating point	0.00001	0.00016	0.00001	4

After the bit-flip mutation is complete, the genetic algorithm checks to make sure that the parameters of produced heuristic search algorithms are within acceptable boundaries. The crossover operation and bit-flip mutation are repeated as necessary to produce a valid offspring.

Each produced offspring is a pair of heuristic search algorithms for solving nested SASSY optimization problems. Each offspring is then asked to search the assigned finalist sample problem. This search is repeated n times, and the score of the heuristic pair, $\overline{U}_{\mathcal{H}_\mathcal{A},\mathcal{H}_Z}$, is computed as follows:

$$\overline{U}_{\mathcal{H}_\mathcal{A},\mathcal{H}_Z} = \frac{1}{n}\sum_{i=1}^{n} U_g(\mathcal{A}_i, Z_i) \qquad (13)$$

where $\mathcal{A}_i$ and Z_i are respectively the best architecture and service selection found in optimization search instance i. In the work presented here, n has been set to 50.

The results for a given offspring are stored in a hash table. If another individual is encountered matching that offspring later in the meta-optimization search, the evaluation of the heuristic pair can be skipped, and $\overline{U}_{\mathcal{H}_\mathcal{A},\mathcal{H}_Z}$ can be recovered from the hash table.

The genetic algorithm continues producing new generations until the heuristic pair evaluation limit is reached (set to 1,000 evaluations in this work). This meta-optimization genetic algorithm was applied to each of the twelve finalist sample problems. The resulting heuristic pairs are shown in Table IV and Table V.

From the results in Tables IV and V, evolutionary programming is clearly the dominant heuristic search algorithm for the service selection search. At the architecture search level, a variety of local search algorithms were found to be optimal on their respective problems. A common feature across all 12 meta-optimization runs are the large values for N_Z. Only problem L generated a heuristic pair with N_Z set to less than 2,000.

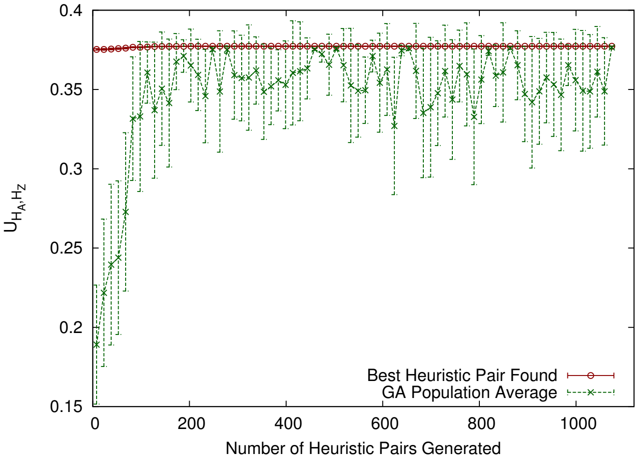


Figure 4. Heuristic pair performance on problem D with 95% CI error bars.

Figures 4 and 5 plot the progress of the meta-optimization search on the finalist sample problems D and F respectively. Due to differences in the environment, the scale of the plots' y-axis differ substantially.

TABLE IV. RESULTING HEURISTIC PAIRS FOR FINALIST PROBLEMS A THROUGH F.

Parameter	problem A	problem B	problem C	problem D	problem E	problem F
arch. search budget, $N_{\mathcal{A}}$	19	19	19	19	19	20
arch. search alg.	beam search	hill-climbing	hill-climbing	beam search	hill-climbing	beam search
arch. search mode	exceeds LL	greedy	opportunistic	no LL req.	greedy	no LL req.
arch. # of filter SSSes	2	6	12	4	3	4
arch. # of filter comp.	2	24	4	5	1	1
arch. beam width	4	N/A	N/A	4	N/A	5
arch. ini. prob.	N/A	N/A	N/A	N/A	N/A	N/A
arch. final prob.	N/A	N/A	N/A	N/A	N/A	N/A
serv. sel. search budget, N_Z	2,475	2,475	2,475	2,475	2,475	2,275
serv. sel. search alg.	evol. prog.	evol. prog.	evo. prog.	evol. prog.	evol. prog.	evol. prog.
serv. sel. par. pop. size	2	1	1	4	1	4
serv. sel. off. pop. size	5	7	2	8	6	4
serv. sel. overlap pop.	true	true	true	true	false	true
serv. sel. ini. step size	4.5	2.5	3.0	4.5	2.5	4.5
serv. sel. adapt. step fact.	1.0	1.5	1.5	3.5	1.5	1.5

TABLE V. RESULTING HEURISTIC PAIRS FOR FINALIST PROBLEMS G THROUGH L.

Parameter	problem G	problem H	problem I	problem J	problem K	problem L
arch. search budget, $N_{\mathcal{A}}$	22	20	19	21	23	32
arch. search alg.	hill-climbing	sim. annealing	hill-climbing	hill-climbing	hill-climbing	hill-climbing
arch. search mode	opportunistic	N/A	opportunistic	greedy	opportunistic	opportunistic
arch. # of filter SSSes	11	N/A	unused	3	12	11
arch. # of filter comp.	3	N/A	unused	1	2	2
arch. beam width	N/A	N/A	N/A	N/A	N/A	N/A
arch. ini. prob.	N/A	0.26	N/A	N/A	N/A	N/A
arch. final prob.	N/A	0.0008	N/A	N/A	N/A	N/A
serv. sel. search budget, N_Z	2,100	2,375	2,500	2,250	2,050	1,475
serv. sel. search alg.	evol. prog.	evol. prog.	evo. prog.	evol. prog.	evol. prog.	evol. prog.
serv. sel. par. pop. size	1	3	3	2	3	3
serv. sel. off. pop. size	4	18	22	6	12	15
serv. sel. overlap pop.	true	true	true	true	true	true
serv. sel. ini. step size	4.5	2.5	3.5	1.0	4.0	3.0
serv. sel. adapt. step fact.	1.0	1.0	2.0	1.0	1.5	1.0

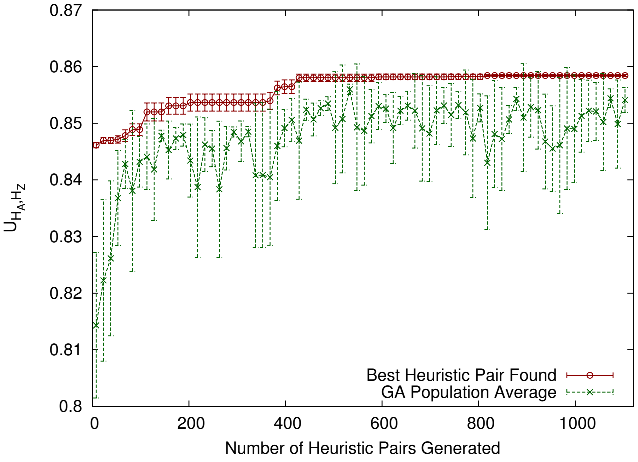


Figure 5. Heuristic pair performance on problem F with 95% CI error bars.

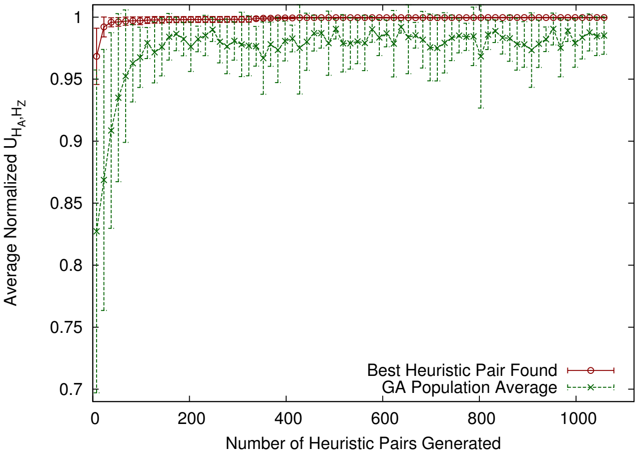


Figure 6. Normalized heuristic pair performance across all problems with 95% CI error bars.

Each of the finalist sample problems has a different y-scale. To gauge the overall convergence of the meta-optimization genetic algorithm, we normalize the search performance against the best U_g found during the entire meta-optimization search. A plot of the normalized convergence can be found in Figure 6.

V. EXPERIMENTAL EVALUATION

After the meta-optimization genetic algorithm produced optimized heuristic algorithm pairs for each of the twelve finalist problems, we tested these twelve heuristic pairs in simulation (see Tables IV and V). This simulation software was originally developed for the experimental evaluation in [3]. As a control, we also tested the beam search/evolutionary programming BS−EP heuristic search algorithm pair from [3] (see Table VI).

TABLE VI. CONTROL HEURISTIC PAIR PARAMETER SETTINGS.

Parameter	control
arch. search budget, $N_{\mathcal{A}}$	63
arch. search alg.	beam search
arch. search mode	no LL req.
arch. # of filter SSSes	5
arch. # of filter comp.	2
arch. beam width	2
arch. ini. prob.	N/A
arch. final prob.	N/A
serv. sel. search budget, $N_{\mathcal{Z}}$	756
serv. sel. search alg.	evol. prog.
serv. sel. par. pop. size	3
serv. sel. off. pop. size	19
serv. sel. overlap pop.	true
serv. sel. ini. step size	3.5
serv. sel. adapt. step fact.	4.5

A. Simulation Parameters

Each simulation commences with the SOA application in a near-optimal architecture that was found in a lengthy, offline heuristic search. The simulation time is divided into discrete intervals called *controller intervals* of duration ϵ time units.

The following actions take place at the end of each controller interval:

- SPs that are active and up will be scheduled to go down t_{fail} time units after they become operational. The time t_{fail} is drawn from an exponential distribution with an average equal to the SP's Mean Time To Failure (MTTF). This exponentially distributed number is rounded up to the closest multiple of ϵ. Thus, at the end of each controller interval, if any SP is scheduled to go down at that time, the SP is flagged as down, and the software system's U_g is computed and recorded.
- For each SP that failed at the end of a controller interval, an exponentially distributed number t_{recover} with average equal to the SP's Mean Time To Repair (MTTR) is selected. The value of t_{recover} is rounded up to the closest multiple of ϵ. Thus, at the end of a controller interval, if any SP is scheduled to recover, the SP is flagged as operational again. The autonomic controller conducts a re-architecting search to see if the new SP can be used to attain a higher U_g.
- Compute the U_g. If it falls below a certain set threshold, initiate rearchitecting.

Separate Mersenne Twister random number streams were used for the generation of simulation events and for heuristic search calculations. The duration of each simulation was 500 ϵ. We conducted 100 simulations for the control heuristic pair and for each of the twelve heuristic algorithm pairs generated by the meta-optimization process.

B. Experimental Results

Each autonomic controller encountered approximately 400 re-architecting events over the course of a single simulation run. Figure 7 shows the distribution of average global utilities in each set of 100 experiments produced by the twelve heuristic pairs and the control. The boxes in this figure show the three population quartiles, while the whiskers show the maximum and minimum.

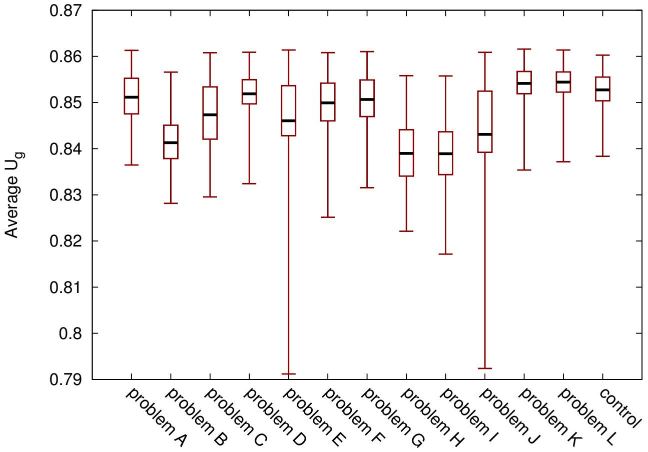


Figure 7. Box plot showing the quartiles of the simulation runs.

The average U_g maintained over the 100 simulations with 95% confidence intervals is presented in Table VII. A visual test of the confidence intervals shows that the heuristic pair generated for problem L performed better than each of the other heuristic pairs except for that generated for problem K. Next, we assess the statistical significance of the results.

TABLE VII. 95% CONFIDENCE INTERVALS FOR AVERAGE GLOBAL UTILITY.

Heuristic Pair	lower bound	mean	upper bound
control	0.8520	0.8527	0.8535
problem A	0.8501	0.8511	0.8522
problem B	0.8403	0.8413	0.8423
problem C	0.8459	0.8473	0.8488
problem D	0.8509	0.8519	0.8529
problem E	0.8436	0.8461	0.8485
problem F	0.8487	0.8499	0.8511
problem G	0.8496	0.8507	0.8518
problem H	0.8376	0.8390	0.8404
problem I	0.8376	0.8389	0.8402
problem J	0.8403	0.8431	0.8459
problem K	0.8533	0.8541	0.8550
problem L	0.8537	0.8544	0.8552

We applied the Tukey-Kramer procedure to the twelve heuristic pairs and to the control heuristic pair with $\alpha = 0.05$ and determined the following:

- The heuristic pair generated by the meta-optimization for problem L (opportunistic hill-climbing/evolutionary programming) was superior to nine of the twelve heuristic pairs generated for the other problems. Results comparing its performance to those generated for problems A, D, K, and the control were inconclusive.
- The heuristic pair generated for problem K was superior to eight of the twelve heuristic pairs generated for the other problems. Results comparing to A, D, G, L, and the control were inconclusive.
- The control pair was superior to half of the generated heuristic pairs; the results comparing to A, D, F, G, K, and L were inconclusive.

To obtain more conclusive results, we reduced the variance caused by the inferior performance of certain heuristic pairs by repeating the test with the top performing heuristic pairs, thereby increasing the granularity of the Tukey-Kramer procedure. When considering just the heuristic pairs generated for problems A, D, K, L, and the control, we found the following:

- The heuristic pair generated for problem L was superior to those generated for problems A and D.
- The heuristic pair generated for problem K was superior to that generated for problem D.

We further reduced the variance to permit comparisons among the top three heuristic pairs: for problem K, for problem L, and the control. We found the following:

- The heuristic pair generated by the meta-optimization for problem L was superior to the control.
- The heuristic pair generated by the meta-optimization for problem K was also superior to the control.

VI. Related Work

Early work in meta-optimization of heuristic search algorithms was performed by Grefenstette [13]. In this work, genetic algorithms (GAs) were used to optimize other GAs. The motivation for this work was similar to ours: a reduction in the human effort required to select appropriate parameters controlling the GA's behavior. Similar to Grefenstette, Keane [14] focuses on meta-optimization of GAs used in multi-peak engineering problems. The GAs are meta-optimized by both GAs and simulated annealing. A more sophisticated approach that focuses on improving GA performance on mixed integer optimization is presented by Bäck in [15]. In this work, the meta-optimization algorithm is a hybrid of evolution strategies and a GA.

In [16], Meissner et al. develop a particle swarm optimization (PSO) meta-optimization technique using a super-particle swarm that manages the parameters of sub-particle swarms with a focus on optimizing neural networks. In his dissertation thesis [17], Pedersen presented a meta-optimization that he applied to PSO and Differential Evolution. His meta-optimizer found simpler algorithms were often more effective.

In [18], Stephenson et al. employ an evolutionary algorithm for meta-optimizing compiler heuristics. Similar to our work here, reducing human effort in tuning heuristics was a primary motivation for this work.

A literature review of software architecture optimization that provides a useful roadmap for comparing features and categorizing work in this field can be found in [19].

In [20], Calinescu et al. present QoSMOS, a system for online performance management of SOA systems. Like SASSY, this system employs utility functions to combine multiple QoS objectives and optimizes the selection of SPs. Unlike SASSY QoSMOS considers the SPs to be white boxes, and it can adjust the configuration parameters and ersource allocations for those white box SPs. Also, QosMOS does not employ architectural patterns for improving QoS. Finally, QoSMOS uses exhaustive search, a technique that cannot be used in near real-time at the scale presented in our paper.

Cardellini et al. devise a framework, MOSES, for optimizing SOA systems in [4]. Similar to SASSY, MOSES uses SP selection and architectural patterns for improving the QoS of a SOA service or application. MOSES adapts the optimization problem such that it can be solved through linear programming (LP) techniques. LP techniques operate well on convex objective functions but are substantially less effective on concave objective functions with multiple optima. The optimization techniques presented in our paper are more effective on concave global utility functions with multiple optima.

Other researchers have investigated using multi-objective optimization techniques to reduce effort and increase the quality of software architecture designs. When the optimization search completes, these systems present human decision makers with a set of Pareto optimal architecture candidates. PerOpteryx, introduced by Koziolek et al. in [21], employs architectural tactics in a multi-objective evolutionary algorithm to expedite the multi-objective search process; later work extends this approach in [22]. Martens et al. present a similar system in [23] that starts quickly by using LP on a simplified version of the problem to prepare a starting population for a multi-objective evolutionary algorithm.

VII. Conclusion

The meta-optimization was successful. Some of the resulting heuristic pairs exceeded even the performance of the control, which had previously been shown to be optimal on a different SASSY application [3], and which performed well in comparison to many of the meta-optimized heuristic pairs in these experiments.

Of the twelve heuristic pairs generated by the meta-optimization, the heuristic pairs produced for problems K and L possessed the largest architecture search budgets (23 and 32 respectively), while the control heuristic pair had an architecture search budget of 63. These settings are likely due to the more challenging nature of problems K and L as compared to A through J. Both the K and L heuristic pairs use opportunistic hill-climbing for the architecture search algorithm; this leverages the architecture search budget by ensuring the search can visit a number of architecture neighborhoods.

For this SASSY application, having an effective architecture search is key to succeeding on the more challenging optimization problems. Those heuristic pairs produced for less challenging problems de-emphasized the architecture search in favor of the service selection search. This provides marginal benefits when solving the easiest problems, but is a significant liability on more challenging problems and can lead to lower global utility values over time.

The relatively wide range in the performance of meta-optimized heuristic pairs highlights the importance of running the meta-optimization on a diverse set of problems, including outliers (both problems K and L were outliers). When performing future meta-optimizations in SASSY, we will consider using a larger set of finalist sample problems to ensure the presence of challenging problems.

Using meta-optimized heuristic pairs on SASSY provides cumulative global utility benefits over time. Furthermore, the generation of the meta-optimized heuristic pairs was automated and required minimal human administration. The meta-optimization process lowered costs by reducing the human effort required to find effective heuristic pairs. Thus, we have achieved better performance at reduced cost.

In future work, the meta-optimization process could be fully automated. This would allow online SASSY meta-controllers in [3] to use the meta-optimization framework presented here. A logical question when considering meta-optimization is: ”What or who will manage the meta-optimization process?” Like the autonomic controller it manages, the meta-controller contains a number of tunable parameters. Has the introduction of the meta-optimization process moved the management overhead to a new component?

Although setting up a meta-optimization process requires some initial effort from human administrators, there is an argument that this effort will be minimal compared to managing the autonomic controller itself. The autonomic controller is closer to the dynamic environment of the managed system than the meta-optimization process. This dynamism can cause problems for an autonomic controller.

However, the immediate environment of the meta-optimization process is more static. The meta-optimization's environment changes only when large changes are made to the autonomic controller (e.g., the introduction of new heuristic search algorithms or a significant evolution of the managed SOA application). Even when such large changes occur, a properly constructed and tested meta-optimization process should be able to weather the change with minimal human intervention. Thus, the meta-optimization process represents a significant step towards developing fully *autonomic* systems.

Finally, we believe the overall meta-optimization approach presented here could be adopted in other self-adaptive, self-optimizing autonomic systems.

References

[1] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” IEEE Computer, vol. 36, no. 1, Jan. 2003, pp. 41–50.

[2] G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani, “A hybrid reinforcement learning approach to autonomic resource allocation,” in Proc. 3rd IEEE International Conference on Autonomic Computing (ICAC ’06), Dublin, Ireland, Jun. 2006, pp. 65–73.

[3] J. M. Ewing and D. A. Menascé, “A meta-controller method for improving run-time self-architecting in soa systems,” in Proceedings of the 5th ACM/SPEC international conference on Performance engineering. ACM, 2014, pp. 173–184.

[4] V. Cardellini, E. Casalicchio, V. Grassi, S. Iannucci, F. Lo Presti, and R. Mirandola, “Moses: A framework for QoS driven runtime adaptation of service-oriented systems,” Software Engineering, IEEE Transactions on, vol. 38, no. 5, 2012, pp. 1138–1159.

[5] M. C. Huebscher and J. A. McCann, “A survey of autonomic computing–degrees, models, and applications,” ACM Computing Surveys, vol. 40, no. 3, Aug. 2008, pp. 1–28.

[6] D. A. Menascé, J. M. Ewing, H. Gomaa, S. Malek, and J. P. Sousa, “A framework for utility-based service oriented design in SASSY,” in Workshop on Software and Performance, San Jose, CA, Jan. 2010, pp. 27–36.

[7] D. A. Menascé, H. Gomaa, S. Malek, and J. Sousa, “Sassy: A framework for self-architecting service-oriented systems,” IEEE Software, vol. 28, no. 6, Nov. 2011, pp. 78–85.

[8] D. A. Menascé, J. P. Sousa, S. Malek, and H. Gomaa, “QoS architectural patterns for self-architecting software systems,” in Proc. 7th International Conference on Autonomic Computing (ICAC ’10), Washington, DC, Jun. 2010, pp. 195–204.

[9] H. Gomaa, K. Hashimoto, M. Kim, S. Malek, and D. A. Menascé, “Software adaptation patterns for service-oriented architectures,” in Proc. 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, Mar. 2010, pp. 462–469.

[10] K. DeJong, Evolutionary Computation. Cambridge, MA: MIT, 2002.

[11] V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith, Eds., Modern Heuristic Search Methods. Hoboken, NJ: Wiley, 1996.

[12] J. Rowe, D. Whitley, L. Barbulescu, and J.-P. Watson, “Properties of gray and binary representations,” Evolutionary Computation, vol. 12, no. 1, 2004, pp. 47–76.

[13] J. J. Grefenstette, “Optimization of control parameters for genetic algorithms,” Systems, Man and Cybernetics, IEEE Transactions on, vol. 16, no. 1, 1986, pp. 122–128.

[14] A. J. Keane, “Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness,” Artificial Intelligence in Engineering, vol. 9, no. 2, 1995, pp. 75–83.

[15] T. Bäck, “Parallel optimization of evolutionary algorithms,” in Parallel Problem Solving from NaturePPSN III. Springer, 1994, pp. 418–427.

[16] M. Meissner, M. Schmuker, and G. Schneider, “Optimized particle swarm optimization (opso) and its application to artificial neural network training,” BMC bioinformatics, vol. 7, no. 1, 2006, p. 125.

[17] M. E. H. Pedersen, “Tuning & simplifying heuristical optimization,” Ph.D. dissertation, University of Southampton, 2010.

[18] M. Stephenson, S. Amarasinghe, M. Martin, and U. O’Reilly, “Meta optimization: Improving compiler heuristics with machine learning,” SIGPLAN Not., vol. 38, no. 5, 2003, pp. 77–90.

[19] A. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya, “Software architecture optimization methods: A systematic literature review,” Software Engineering, IEEE Transactions on, vol. 39, no. 5, 2013, pp. 658–683.

[20] R. Calinescu, L. Grunske, M. Kwiatkowska, R. Mirandola, and G. Tamburrelli, “Dynamic QoS management and optimization in service-based systems,” Software Engineering, IEEE Transactions on, vol. 37, no. 3, 2011, pp. 387–409.

[21] A. Koziolek, H. Koziolek, and R. Reussner, “Peropteryx: automated application of tactics in multi-objective software architecture optimization,” in QoSA-ISARCS ’11. New York, NY, USA: ACM, 2011, pp. 33–42.

[22] A. Koziolek, D. Ardagna, and R. Mirandola, “Hybrid multi-attribute qos optimization in component based software systems,” Journal of Systems and Software, vol. 86, no. 10, 2013, pp. 2542–2558.

[23] A. Martens, D. Ardagna, H. Koziolek, R. Mirandola, and R. Reussner, “A hybrid approach for multi-attribute QoS optimisation in component based software systems,” in Research into Practice–Reality and Gaps, ser. LNCS, G. Heineman, J. Kofron, and F. Plasil, Eds. Springer Berlin Heidelberg, 2010, vol. 6093, pp. 84–101.

Efficient Management of Cooling Systems in Green Datacenters

Ibrahim Safieddine, Noël de Palma
ERODS - LIG
Université Grenoble Alpes
Grenoble, France
email: {ibrahim.safieddine, noel.de-palma}@imag.fr

***Abstract*—Next-generation green datacenters were designed for optimized Power Usage Effectiveness (PUE), which is the ratio of the total power consumption of the datacenter over the computing consumption. Continuous improvement of these datacenters target the reduction of PUE in accordance with the servers' load, to approach the minimal target, i.e., PUE = 1. Datacenters must ensure in the same time a very high level of availability of resources and good management of failures, thanks to redundant equipments, while optimizing power consumption and cooling costs and reducing the environmental footprint. A datacenter consists of computing, power distribution and cooling parts. The cooling part represents the main cost that significantly increases the power bill and consequently the PUE. A datacenter can be cooled using heterogeneous cooling systems for redundancy in the case of failure. These systems have a variable consumption depending on the load and on external parameters, e.g., weather and external temperature. This paper presents an efficient cooling manager, which aims to minimize the PUE while satisfying the Service Level Agreement (SLA), by reducing the power consumption of the datacenter and opting for the most efficient cooling system according to climate conditions and by limiting temperature variations and cooling mode transitions. Our system uses an overview of all datacenter parts to provide an optimal decision that complies with regulations when using natural resources, e.g., groundwater.**

Keywords*—Power Usage Effectiveness (PUE); cooling systems; green datacenter; autonomic computing; service level agreement (SLA)

I. INTRODUCTION

The datacenter market is growing at a rate of 15% per year globally [1]. Green computing, which aims to reduce energy consumption and the related greenhouse effect, remains a priority for datacenter managers due to the increase in the price per kWh, e.g, France's electricity bill is expected to increase by 50% in 2020 [2]. As servers become smaller without necessary consuming less energy, datacenters which have more and more servers generate more heat, and the cooling power needed will grow. The concentration of computing power per m^2 has grown very quickly over the last 10 years from 15 to 30 kW per rack, which results in significant heat dissipation and thus a higher cost for cooling.

Knowing that the electricity bill is the main operational charge in a datacenter, the new challenge of datacenters is the mastery of electrical distribution, the choice of the best cooling technologies, e.g., air-cooled or water-air, and its optimization for better performance. The intelligence in a datacenter relies on sensor networks that provide real-time measurements and robust industrial automation control, to find the best operating point.

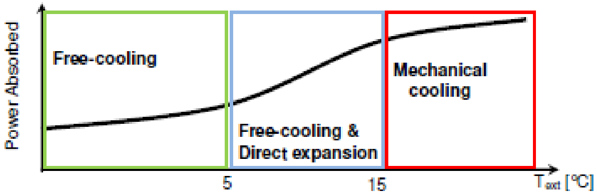


Figure 1. Hybrid cooling consumption.

The objective of datacenters is to ensure near 100% facilities availability, through redundant components which keeps the systems up even in case of the failure of an active element or during maintenance. Hence, the importance of optimizing the very complex cooling system as a whole, by analyzing a large number of parameters (ambient temperature, humidity, weather forecast, servers load, etc.) and having an overview of all datacenter equipments for a more efficient management of cooling. This will impact the overall consumption in the datacenter and the cost of operation, and thus improves the PUE and reduce costs.

There are many efficient cooling systems [3] [4] [5] [6]. Free cooling [3] is an economic method that uses low external air temperature and less power to cool water instead of mechanical refrigeration. Hybrid cooling [4] incorporates two cooling modes, free and electrical cooling, with an internal system that switches between these modes depending on the outside air temperature. Figure 1 shows the variation of the consumption of an hybrid cooling system with the external temperature [7].

Another cooling system that uses groundwater [5] and rivers' [6] water is a very economical cooling system. The system works by pumping cool water through a heat exchanger and then re-injecting the heated water back into the source, resulting in no net loss of groundwater. It uses the cold water in an open loop to cool the internal circuit water.

In this paper, we propose an autonomic solution that efficiently manages and optimizes the choice of the cooling system in a high available green datacenter while satisfying the SLA. Our algorithm can manage many heterogeneous cooling systems with different cooling capacities to minimize the cooling power consumption and can use multiple cooling systems simultaneously for better efficiency. To improve our solution, we correlated the datacenter internal measurements with other indicators such as external temperatures and weather forecast.

We used a global vision on all datacenter layers, FaaS (Facilities as a Service)-IaaS (Infrastructure as a Service)-PaaS (Platform as a service)-SaaS (Software as a Service), to avoid conflicted decisions and process in a better and faster way. We validate our algorithm on a real architecture and real measurements from the green datacenter of Eolas[1] [8] that can host more than 13,000 servers. We then demonstrate that this solution can be scaled to include other cooling systems and can be adapted on more complicated environments. The rest of this paper is organized as follows. In Section II, we present the state of the art on cooling system optimization algorithms. In Section III, we describe our algorithm for optimizing the management of cooling systems. In Section IV, we present the architecture of a redundant green datacenter on which our work is based. Finally, in Section V, we summarize the conclusions and future perspectives of this work.

II. Related work

In order to reduce the power consumption of cooling systems, several solutions are proposed. Some solutions are based on varying water or air temperature in the servers' room to reduce the cooling power. This solution can be coupled with servers consolidation to reduce the dissipation of heat. Other global approaches aims to reduce the consumption of the cooling infrastructure by switching between free and electrical cooling according to external temperatures.

Shaoming et al. [9] discuss the impact of server consolidation and the variation of the cold aisle temperature in a servers room, on the cooling consumption of the datacenter. They focus on optimization of cooling consumption and maintenance costs. Increasing the temperature by 1°C can reduce energy cooling consumption by 2 to 5%, however high temperature reduce the reliability of electronic components, and increases the cost of hardware maintenance of CPU and Memory. In the same way, the consolidation that consists on cycles of start/stop servers, decrease the cooling costs but in the same time decreases the hard-disks lifetime.

While the previous paper focuses on increasing inlet temperature and servers consolidation in servers room, Jungsoo et al. [3] use servers consolidation and exploit time-varying servers workload and external climate conditions in order to reduce the power consumption of the entire datacenter. This system is based on the maximum usage of the free-cooling and preventing frequent cooling transition.

Ratnesh and al. [10] propose a framework for dynamic thermal management based on asymmetric workload placement that can promote uniform temperature distribution that reduces local hot spots, quickly responds to thermal emergencies, reduces energy consumption costs, reduces initial cooling system capital costs and improves equipment reliability. This framework is not related to cooling systems capacity.

III. Implementation

The basic automatons implemented in most of the cooling systems are very reliable and can manage efficiently the availability of cooling in case of failure. However, they are limited to the predefined priority established by the Data Center Manager (DCM) and each automaton can manage a limited number of cooling systems and works independently from other datacenter layers. In our implementation, we used the requested cooling capacity and the external weather conditions to improve the global efficiency of the cooling system and reduce the datacenter consumption. Using weather forecast, we can limit the transition between cooling systems when the external temperature is constant or when the temperature decrease while using free-cooling. In hybrid cooling, we can predict the cooling system that will be used (free-cooling, mixed or chiller cooling) depending on the temperature forecasts before starting the system. In addition, the usage of the groundwater cooling is highly regulated: the water flow and the yearly water volume is limited by the law. Our algorithm proposes a better way to manage the groundwater cooling system based on external temperature and servers load history throughout the year. Knowing that it is possible to start several cooling system simultaneously, it is important to reduce the number of active cooling systems to reduce the consumption.

In order to prevent damage to cooling system by repetitive start/stop cycles, we defined a minimal period between two transitions, based on the systems data manual.

To minimize the electricity cost and then reduce the PUE, we setup the cooling power requirement of the datacenter model which estimate the impact of servers consumption on the cooling power needed, as presented in (1):

$$\sum C_P(kW) = 0.9 * \sum S_C(kW) \tag{1}$$

where C_P is the cooling power needed in kW and S_C the total electrical consumption of servers in kW. Almost, all the power consumed by the server is transformed to heat.
For the optimization problem, we used a linear optimization program, where the goal is to minimize the cooling electrical power consumption linear function while respecting operation constraints. Linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints.

Equation (2) represents the cost in € to produce one kW of cooling. P_{elec} and k€ respectively denote the power consumed by a cooling system to produce 1kW of cooling and kWh Billing rate which comes to 6 cents/kWh in France.

$$C = P_{elec} * k\text{€} \tag{2}$$

The global optimization program can be presented as follows:

$$Min \sum_{i=0}^{n} a_i.C_i \tag{3}$$

[1] Eolas (www.eolas.fr) is a French company based in Grenoble, part of the international Consulting and System Integration (CSI) company Business & Decision

$$\text{Subject to}: \begin{cases} \sum_{i=0}^{n} a_i >= C_P & i \in 0, n \\ \sum Cm_i \in \{\text{possible cooling modes}\} \end{cases} \tag{4}$$

Equation (3) represents the objective function of the linear optimization program where a_i represent the amount of cooling produced by the cooling system i in kW and C_i the cost in € of 1kW of cooling using the cooling system i. n represents the maximum number of cooling systems that can be started simultaneously. We look to minimize the total consumption of started cooling systems and limit the number of started systems. While minimizing the cooling systems consumption, we minimize the datacenter consumption and then reduce the PUE. Equation (4) represents the constraints of the optimization problem. We need to satisfy the SLA by delivering as match cooling power as needed, so the total cooling power generated must be ideally equal to the needed cooling power. Cm_i is the cooling mode of the cooling system i. The possible cooling modes list is limited by many parameters as the external temperature or a high temperature of the groundwater.

Weather forecasts are used to estimate at each future period the best cooling system to be used and then limit cooling mode transitions with temperature variations.

IV. Evaluation

In this section, we present the Eolas datacenter cooling architecture used to evaluate our algorithm. The green datacenter of Eolas is Tier 4, and therefore, designed to host mission critical computer systems, with fully redundant subsystems. Figure 2 shows that 3 principal cooling systems are used for maximum redundancy. All cooling equipment is independently dual-powered, including chillers, ventilation and air-conditioning systems. Those systems are heterogeneous: groundwater cooling (with two independent pumps), hybrid cooling and chiller cooling. Eolas uses an ultimate cooling source: city water. When all cooling systems fall down or in case of power failure, the city water (having a temperature of 12 - 14 °C) can be used to cool up to 3000 servers. This ultimate cooling source is very expensive and increase the WUE (Water Usage Effectiveness) of the datacenter.

Stopping the cooling system can be dramatic for this datacenter, i.e., every minute, the room temperature increases by 1°C, knowing that the hot aisle is fixed at 35°C, thus the room temperature may reach 60°C in just 25 minutes.

Actually, the transition between cooling modes is done manually in this datacenter. Using our algorithm, the transition is full automatic and based on several external sensors for more efficiency.

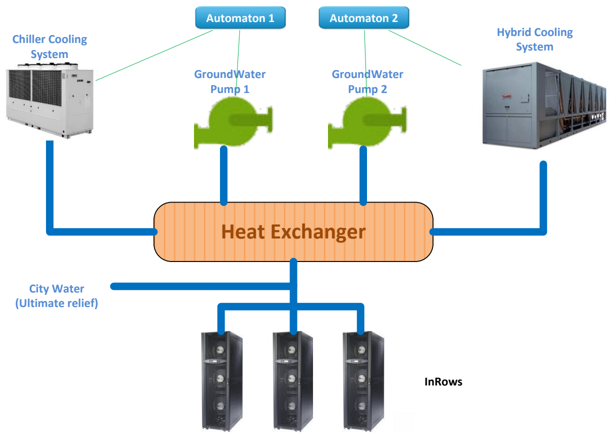


Figure 2. The global cooling architecture.

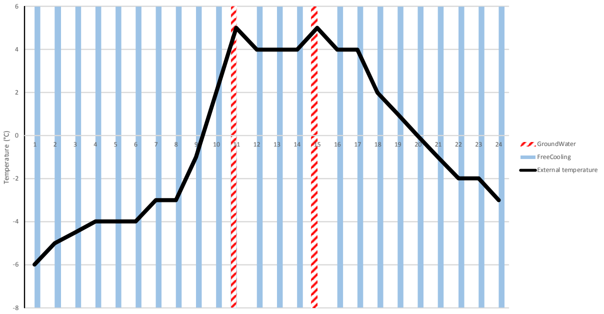


Figure 3. Cooling modes transitions in May in normal mode.

Figure 3 shows the automatic cooling mode transition between groundwater and Free cooling systems, using our optimisation algorithm, without weather forecasts in a day of May. When the temperature is too low, the Free cooling system is very efficient, i.e., his Energy Efficiency Rating (EER) is two high. When the temperature increases, the EER start decreasing and the groundwater system became more efficient. In this example, the system switches to groundwater cooling mode when the external temperature reaches 5°C. This transition is useless since the duration before returning to the Free cooling mode is too short. Using weather forecasts, the datacenter will be cooled using free cooling all the day, with no cooling systems transition.

First results for a small example illustrate the potential for a coordinated control strategy to achieve better energy management than traditional industrial automatons that control the cooling systems separately. We can save up to 38% of cooling power using our algorithm.

V. Conclusion and Future Work

In this paper, we proposed an autonomic optimization system for heterogeneous cooling systems in a Tier 4 green datacenter. Our algorithm, connects to existing automatons and all datacenter sensors and uses external conditions and weather forecast to choose the best cooling systems combination to reduce the overall power consumption in the datacenter and limiting cooling mode transitions. We experimented this work using real data, collected from the Eolas green datacenter at Grenoble, France. As future enhancements of our solution, we intend to integrate a predictive model of the datacenter activity to predict the future cooling power needs and minimize the transition between cooling systems accordingly, in order to reduce systems failure.

Acknowledgment

This research is partially supported by ANR INFRA under a grant for the project Ctrl-Green (ANR-11-INFR 012-11) [11] and by the Datalyse project [12].

References

[1] J. G. Koomey, Growth in data center electricity use 2005 to 2010, Analytics Press, october, 2011.

[2] La commission d’enquête sur le coût réel de l’eĺectricité, SENAT - France, July 2012.

[3] J. Kim, M. Ruggiero, and D. Atienza, “Free cooling-aware dynamic power management for green datacenters,” High Performance Computing and Simulation (HPCS), 2012 International Conference on, 2012, pp. 140-146.

[4] M. K. Patterson, D. Atwood, and J. G. Miner, “Evaluation of air-side economizer use in a compute-intensive data center,” ASME 2009 InterPACK Conference, Vol. 2, pp. 1009-1014, July 2009.

[5] M. P. 1, “Sustainable groundwater-source cooling systems for buildings,” Proceedings of the ICE - Engineering Sustainability, Vol. 161, Iss. 2, June 2008 , pp 123 133.

[6] P. Dalin, “Free cooling/natural cooling is crucial for a successful district cooling development,” Teaming Up for renewable heating and cooling, April 2012.

[7] Engineering Data Manual - Aquaflair, Uniflair, October 2011.

[8] (2015, March) Eolas. [Online]. Available: http://www.businessdecision-eolas.com/544-green-it.htm

[9] S. Chen, Y. Hu, and L. Peng, “Optimization of electricity and server maintenance costs in hybrid cooling data centers,” 2013 IEEE Sixth International Conference on Cloud Computing, 2008, pp. 526-533.

[10] R. Sharma, C. Bash, C. Patel, and R. Friedrich, “Balance of power: Dynamic thermal management for internet data centers,” Internet Computing, IEEE (Vol.9 , Iss. 1). Feb 2005, pp. 42-49.

[11] (2015, March) Ctrlgreen. [Online]. Available: http://www.ctrlgreen.org/

[12] (2015, March) Datalyse. [Online]. Available: http://www.datalyse.fr/

On the Adoption of Multi-Agent Systems for the Development of Industrial Control Networks: A Case Study

Hosny A. Abbas, Mohammed H. Amin
Department of Electrical Engineering
Assiut Faculty of Engineering
Assiut, Egypt
Email: {hosnyabbas,mhamin}@aun.edu.eg

Samir I. Shaheen
Department of Computer Engineering
Cairo Faculty of Engineering
Giza, Egypt
Email: sshaheen@eng.cu.edu.eg

***Abstract*—Multi-Agent Systems (MAS) are adopted and tested with many complex and critical industrial applications, which are required to be adaptive, scalable, context-aware, and include real-time constraints. Industrial Control Networks (ICN) are examples of these applications. An ICN is considered a system that contains a variety of interconnected industrial equipments, such as physical control processes, control systems, computers, and communication networks. It is built to supervise and control industrial processes. This paper presents a development case study on building a multi-layered agent-based ICN in which agents cooperate to provide an effective supervision and control of a set of control processes, basically controlled by a set of legacy control systems with limited computing capabilities. The proposed ICN is designed to add an intelligent layer on top of legacy control systems to compensate their limited capabilities using a cost-effective agent-based approach, and also to provide global synchronization and safety plans. It is tested and evaluated within a simulation environment. The main conclusion of this research is that agents and MAS can provide an effective, flexible, and cost-effective solution to handle the emerged limitations of legacy control systems if they are properly integrated with these systems.**

Keywords-agent-based applications; industrial control networks; real-time monitoring; supervisory control; agents cooperation.

I. INTRODUCTION

ICN is a general term that encompasses several types of control systems used in industrial production, and often found in the industrial sectors and critical infrastructures, it includes: Programmable Logic Controller (PLC), Distributed Control System (DCS), and Supervisory Control and Data Acquisition (SCADA). They are used in industrial production for controlling equipment or a machine [23]. Nowadays, ICN have experienced the most radical changes since the European industrial revolution. These changes include globalization, decentralization, distribution, openness, and increasing application of Information Technologies (IT). Furthermore, their implementations have migrated from custom hardware and software to standard hardware and software platforms. This evolution of industrial systems has led to reduced development, operational, and maintenance costs as well as providing executive management with real-time information that can be used to support planning, supervision, and decision making. On the other hand, this transformation resulted in the need to adopt new software approaches and styles to handle the challenges of these systems, which are mainly related to quality attributes [1].

Conventional software engineering approaches and tools, such as reported in [24], have proven to have limited capabilities to deal simultaneously with many quality attributes. According to Serugendo et al. [2], the complexity of the near future and even present applications can be characterized as a combination of aspects such as the great number of components taking part in the applications, the knowledge and control have to be distributed, the presence of non-linear processes in the system, the fact that the system is more and more often open, its environment dynamic and the interactions unpredictable.

One of the new software engineering architectural styles is the agent-based approach. MAS are one of the most representatives among artificial systems dealing with complexity and distribution [3][4]. They are seen as a major trend in R&D, mainly related to artificial intelligence and distributed computing techniques, and they have attracted attention in many application domains where difficult and inherently distributed problems have to be tackled [5]. A multi-agent system consists of a set of interacting autonomous agents in a common environment in order to solve a common, coherent task. MAS are often relying on the delegation of goals and tasks among autonomous software agents, which can interact and collaborate with each others to achieve common goals [2].

The main research problem addressed here is the integration of agent technology and legacy systems. In the context of industrial computing, a legacy system can be described as an obsolete computer system that may still be in use because its data cannot be changed to newer or standard formats, or its application programs cannot be upgraded. Consider an old small factory contains a control process for producing something (i.e., chemical process) and as a result of the new market demands, new requirements imposed on the factory owner. The owner will find himself compelled to update his factory to handle the new market demands, which can be related to the product quality or produced quantity. The main challenge that will face the owner is the total update cost. If the owner asked the control system vendor to provide an update to the old legacy system he will be surprised by the high estimated cost for the required system update. In this paper and as industrial software developers, we introduce an effective solution for the owner to update his control system network with low cost by the integration of agent technology and legacy control systems. Agents can add a higher level computing layer(s) to the existing system to compensate its limitations. For instance, an agent can be assigned to a control system (i.e., PLC) to provide it with higher level control algorithms and safety plans. For example, if the legacy PLC was not designed to provide Proportional-Integral-Derivative (PID) controller algorithm, this algorithm can be embedded inside a higher layer agent. The connection between the PLC and the agent can be established using a proper interface as shown in Figure 1 and as will be demonstrated later.

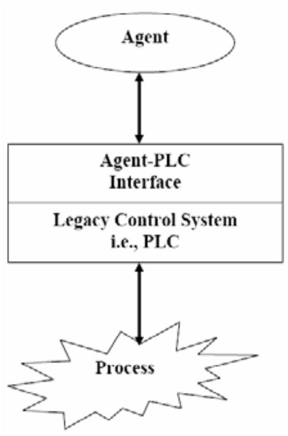


Figure 1. An agent is assigned to a legacy PLC.

In other cases, the concerned factory may contain more than one control system (PLC) and in this case each PLC can be associated with an agent then the agents can cooperate together to provide a type of global synchronization for the underlying control systems. More agents' layers can be added vertically for other purposes for example another top layer can contain remote/local operator agents for providing real-time monitoring to the operators.

This paper presents an approach for building a multi-layered agent-based ICN in which agents cooperate to provide an effective supervision and control of a set of control processes basically controlled by a set of legacy control systems with limited computing capabilities, and to add an intelligent layer on top of basic control systems, in addition to providing a global synchronization and safety plans. The remaining of the paper is organized as follows: Section 2 explores the related work. Section 3 provides a general overview of the proposed ICN. Section 4 presents the development approach of the proposed ICN including a description for each development phase. Section 5 concludes the paper and highlights future work.

II. RELATED WORK

Traditionally, the developers of industrial software applications exploited the widely spread enterprise network, the Internet, to develop efficient web-based industrial applications [25][26]. But as time goes, they discovered that the web technologies, such as web servers and Hypertext Transfer Protocol (HTTP) protocol, still have some limitations related responsiveness, robustness, scalability, adaptability, etc. Moreover, with these technologies the developers are not able to handle simultaneously many quality attributes in one project. A promising solution seems to brighten; it is multi-agent systems. MAS are considered now as a promising solution for handling modern software applications especially industrial applications such as factory automation, supervisory control, real-time monitoring, safety applications, smart grids, home automation and so on. Unfortunately, agent technology generally is not widespread in modern industry (especially in process automation) because of the gap found between agents' theories and industrial applications requirements such as real-time constraints. Some researchers tried to reduce this gap and they adopted the agent-based approach to supervise and control industrial control processes.

In the industrial research, there are many researchers addressed the adoption and deployment of agents and multi-agent systems for industrial purposes. For instance, Metzger and Polakow [27] concluded that the agent technology is particularly popular in the manufacturing domain, while the applications in other domains of industrial control are scarce. They related their conclusions to the lack of the technology support on the part of control instrumentation vendors. In manufacturing automation, the process consists of discrete and countable components and actions. The natural approach is to assign the software agents to each of the components and each of the actions performed. On the other hand, the process automation deals with the continuous physical phenomena, such as chemical reactions. When a process automation system is designed, the phenomena are represented as mathematical models, for which control algorithms are chosen in order to keep the process parameters within a desired range. Therefore, in a single continuous control loop, there is not much place for any additional computational techniques, including the agent technology. Other surveys and reviews such as [9][28][29][30] arrived at the same conclusions.

On the other hand, other researchers developed and implemented many valuable and feasible agent-based industrial applications. For instance, Diaconescu and Spirleanu [7] presented a concrete way of linking a multi-agent system with the equipment (i.e., PLC, DCS, SCADA, and Human Machine Interface (HMI)) comprised into a distributed industrial control system based on agents, using Open Process Control (OPC) servers [13]. Their research concerned with the application of agent technology for monitoring, collection and archiving data of a manufacturing process in the automotive industry. The contributions of the authors are mainly directed to achieve the connection between Java Agent Development (JADE) framework [8] and OPC server but they did not exploit the advanced features provided by JADE, such as ontology support, agents’ cooperation, advanced interaction protocols which are very important especially for open and large scale systems. Pereira et al. [9] discussed the current challenges of the deployment of MAS in the context of industrial applications, mainly focusing the integration of agents with physical equipment and the ability to run agents directly in industrial or low cost controllers. To support their claims the authors provided an experimental MAS solution for a smart grid case study. The authors’ main concern was how to integrate agents with physical equipment. Rupare et al. [10] presented an automated grinding media charging system incorporating a multi-agent system developed in JADE too.

In short, modern industrial applications are badly in need of adopting agents and multi-agent systems as new modeling paradigms to handle their challenges such as scalability, robustness, flexibility, etc. Researchers should continue developing practical industrial projects and applications able to satisfy the applications real-time constraints. This research can be considered as a step towards achieving this goal. It is a step towards building an open and large scale industrial control networks comprises variety of components and equipments work together in and efficient and effective way and concern both real-time supervisory and control activities. Unlike other similar work, the proposed ICN follow an ad hoc methodology which divides the development process to steps easy to follow, understand, and implement.

III. THE PROPOSED ICN OVERVIEW

Regarding conventional software engineering, an ICN is considered as a distributed system. Burmakin et al. [11] described a distributed system to be the system in which the entities are distributed physically and/or logically, the entities are essentially heterogeneous, cross-communication and co-operation between the entities and their environment are key features, and the entities act as a unity to achieve a common goal. From this description of distributed systems, we consider an ICN as a distributed system having all these features and in which resources are shared and the logic of the system is distributed among its components.

Galloway et al. [12] pointed out that in almost every situation that requires machinery to be monitored and controlled an industrial control network will be installed in some form. The proposed ICN consists of four layers, physical control processes, basic control systems, control agents, and remote supervisory agents, respectively from down to top. Figure 2 shows the original ICN with the legacy control systems and Figure 3 presents the proposed updated ICN architecture. Note that the proposed ICN is hypothetical but can easily realized and built on top of a working control processes. In the rest of this section, we describe each of these layers (for the updated ICN) in bottom-up order and show how each layer is interfaced with its top and its bottom layers. *Layer 0* (the bottom layer) in the proposed agent-based ICN is the physical control processes layer; it contains the physical industrial processes, such as control processes, electricity generation, food and beverage processing, transportation, water distribution, waste water disposal and chemical refinement including oil and gas. A control process is controlled directly by control systems (its top layer) such as PLC or DCS.

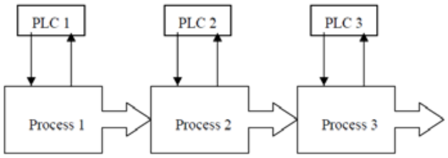


Figure 2. Original Industrial network before update.

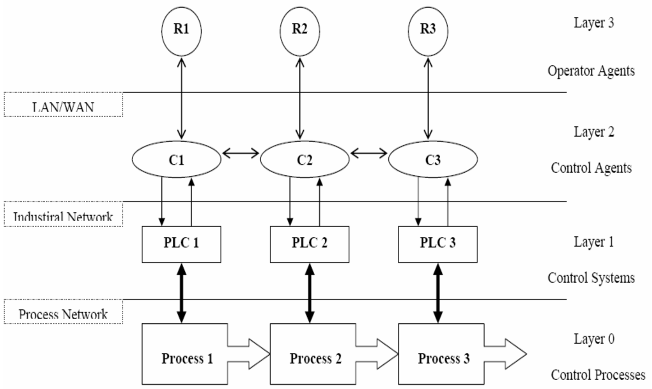


Figure 3. The Proposed layered agent-based ICN Architecture.

Layer1 contains the basic control systems, a control system is a device, or set of devices, that manages, commands, directs or regulates the behavior of other device(s) or system(s). It is a small computer called PLC. The PLC connects to all the electrical sensors, devices, instruments in the industrial process and according to their states; it changes the output states to modify the current state of the industrial process according to a predefined algorithms. *Layer 2* designed to contain control agents used to control, supervise, and synchronize the lower layer control systems. These agents cooperate together by exchanging messages to guarantee the safe, effective, and efficient running of the complete system. Each agent in this layer has an associated PLC and for the sake of control systems interoperability it connects to its control system by OPC protocol [13]. *Layer 3* (the top layer) designed to contain operator local/remote agents. They have user friendly graphical user interfaces (GUIs) to present the system process data in a proper way (i.e., text, graphics, animation, etc.). The local/remote supervisory agents communicate with control and supervisory agents by Agent Communication Language (ACL) [14]. Moreover, the system agents can be connected through a Local Area Network (LAN) or a Wide Area Network (WAN), such as the Internet. The contributions of this research can be summarized as follows:

1. Showing how flexible and straightforward the adoption agent technology for developing feasible and cost-effective industrial networks and compensating the limitations of legacy control systems.
2. Providing real-time monitoring and supervision not only in the local site but also remotely i.e., through the internet.
3. Providing operator support such as checking the validity (i.e., process variables ranges) of operator setpoints, alarm service, and trend or historical process data analysis.
4. Providing higher level control, for example by embedding a PID controller algorithm or similar mathematical algorithms (i.e., interpolation or extrapolation algorithms) inside a control agent.
5. Providing global synchronization among many distributed legacy control systems.

IV. DEVELOPMENT APPROACH

The development life cycle of the proposed ICN comprises four phases: analysis, design, implementation, and evaluation phases. The four development phases are presented in the next subsections. The adopted development approach is an *ad hoc* approach derived from [15] in which a general methodology for JADE applications development was proposed and covered only the analysis and design phases of the development life cycle. Therefore, it will be necessary to augment this methodology with implementation and evaluation phases.

A. Analysis Phase

The first step in the analysis phase is to capture the functional activities of the system-to-be and present these activities in text or graph. One familiar way to do this is by the adoption of *use cases*. Each use case describes a required functional scenario in the system. The use cases have a standard specification included in the Unified Modeling Language (UML) [16], based on the required specification, a complete use case diagram for the proposed ICN is created and is shown in Figure 4. As shown in the figure, the system has two actors communicate through a multi-agent system. The first actor is the human operator who remotely supervises and controls the control process using a remote agent with a friendly Graphical User Interface (GUI). The second actor is the OPC server which can be considered as an active actor because it has the ability to initiate a call back connection with the MAS to provide the MAS with the changed process data. The use case diagram shows that the desired system not only provides a real-time monitoring and operator setpoints' handling services, but also it provides a higher level control on top of underlying legacy control systems. For instance, the system provides a global synchronization among the underlying legacy control systems, checks the validity of operator setpoints submitted through the remote agent GUI (i.e., by checking their valid ranges), sends the proper setpoints to the process control system, and handles process data change events. The next step is to identify agents' types and the number of each agent instances.

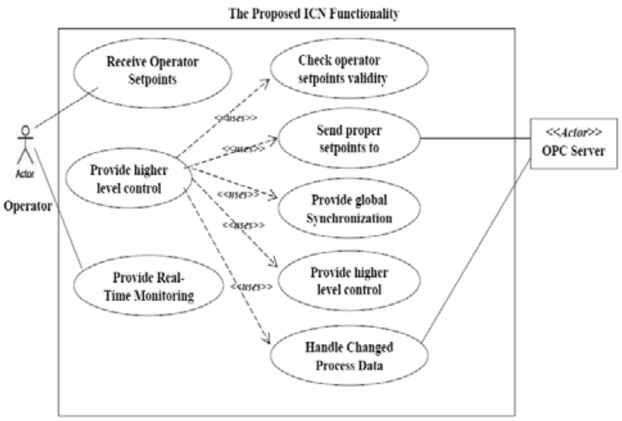

The Proposed ICN Functionality
Receive Operator Setpoints
Check operator setpoints validity
<<uses>>
<<uses>>
Send proper setpoints to
<<Actor>>
OPC Server
Actor
Operator
Provide higher level control
<<uses>>
Provide global Synchronization
<<uses>>
Provide Real-Time Monitoring
<<uses>>
Provide higher level control
Handle Changed Process Data

Figure 4. Use case diagram for the proposed ICN functionality.

Figure 5 presents a final *agent diagram* illustrating the required agent types of the system-to-be. As shown in the figure, the system-to-be comprises only two agent types, the operator agent and the control/supervisory agent, in addition to JADE platform agents such as the directory facilitator agent (DF), which provides the yellow page service to the system-to-be agents. The system may contain more than one control agent (3 in this case study), each of them is associated to a process control system (i.e., PLC).

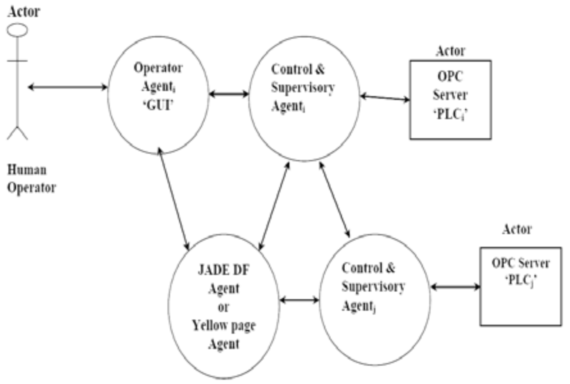

Actor
Human Operator
Operator Agent$_i$ 'GUI'
Control & Supervisory Agent$_i$
Actor
OPC Server 'PLC$_i$'
JADE DF Agent or Yellow page Agent
Control & Supervisory Agent$_j$
Actor
OPC Server 'PLC$_j$'

Figure 5. The agent types diagram with acquaintance relations represented by arrows.

The next step is to identify the responsibilities of each agent in the system; that can be done using the *responsibility table*. Table 1 shows the responsibilities of the system-to-be agents derived from the previously created use case diagram and agent type diagram. The internal functionality of an agent (internal behaviors) is shown in regular and the interaction protocols among agents are shown in *italic*. The existence of interaction protocols demonstrates how the system agents cooperate to achieve the global system goals.

TABLE I. THE PROPOSED ICN RESPONSIBILITY TABLE

Agent Type	Responsibilities
Operator Agent	1. *Discovers Control Agents (search process services)* 2. *Subscribe to a control agent for real-time process data* 3. *Receive real-time process data from a Control Agent* 4. Presents real-time process data to the operator 5. Receives Operator setpoints 6. *Sends operator setpoints to control Agents* 7. *Receives notifications from control agents* 8. presents notifications to operator
Control Agent	1. *Register services with DF* 2. *Subscribe to DF to be notified when control agents register their services* 3. *Handle subscription requests from other control agents* 4. *Subscribe to other control agents for real-time cross process data* 5. *Receives cross process data from other control agents* 6. *Handle subscription requests from operator agents* 7. Receives changed process data from OPC server 8. *Receives Operator setpoints* 9. Checks the suitability of operator setpoints 10. Provides higher level control algorithms 11. Provides global (inter-control agents) synchronization 12. Sends suitable setpoints to its assigned PLC

B. Design Phase

The design phase concerns the transfer from the problem space (analysis phase) to the solution space. It aims to specify the software solution to the problem. It was decided to implement the proposed agent-based ICN using JADE platform, which is a FIPA (Foundation of Intelligent Physical Agents)-compliant agent development platform and is implemented in Java programming language [17]. Therefore, the design phase target is to map the analysis phase artifacts to JADE constructs. Firstly, it is required to classify the system agents' responsibilities shown in the responsibility tables and identify which of them is suitable to be transformed to a JADE *interaction protocol* and which of them is considered as an *internal agent behavior*. Table 2 presents this classification process for the operator agent and similarly Table 3 presents it for the control agent. Another important design issue is related to *agent-resources* interaction. Only the control agents have interaction with non-agent resources, i.e., OPC servers. It is required to find a way to realize agent-OPC interactions. To establish a connection between a JADE agent and an OPC server, a Java-COM bridge or adapter is required. Fortunately, there are many Java-OPC adapters and bridges in the literature, some of them are commercial and some are free source. *JEasyOPC Client* [18] is an example of these bridges; it is a Java OPC client that is now greatly enhanced. It uses a JNI layer coded in Delphi. The current version supports both OPC DA 2.0 and OPC DA 3.0.

The next step in the design phase is to create the application domain *ontology*. Ontology is a set of concepts, predicates and agent actions referring to a given domain. The proposed ontology contains Three main concepts (*ControlProcess*, *Variable*, and *Alarm*), three actions (*SetVariable*, *GetVariable*, *LocateVariable*), and eight predicates:

IsHigh(Variable) - IsLow(Variable)- IsLocal(Variable) - IsLocatedin(Variable,Process)- IsVariable(Variable)- IsControlProcess(ControlProcess)- ListOfVariables(List)- ListOfAlarms(List)

A concept is a complex structure defined by a template specified in terms of a name and a set of slots whose values must be of a given type. A predicate is a relation between domain concepts and its value can be true or false. An agent action is a function the agent is required to perform. The ontology components require a content language to be manipulated and exchanged among agents. A content language is the tool that a message receiver used to decode or parse the message to extract specific information; therefore, the system agents need to agree on a certain content language to understand each other. For the sake of openness and interoperability, the FIPA-SL content language is used in the proposed ICN application. As an example for illustrating the JADE support of ontology and content languages consider these examples:

1. A remote Agent (R1) sends a request message to a control agent (C1) contains a request to write a process variable to a control process as a setpoint:

```
(( action
(agent-identifier :name c1@SCADA :addresses (sequence
http://scada:7778/acc))
(SetVariable :variableAddress s7:[LOCALSERVER]db1,w26
:value 334.0)
))
```

2. The control agent (C1) validates the remote agent setpoint and send an inform message to the remote agent telling it if its action request is carried our or not, the message contains and alarm concept contains the request result as follows:

```
((ListOfAlarms
(sequence (Alarm :destination (agent-identifier
:name R1@SCADA
:addresses (sequence http://scada:7778/acc))
:priority 2
:text "Tue Sep 23 08:34:11 2014 |'PLC1Variable4' New SP
(334.0) was forwarded to control process PLC1"
:var (Variable :lowLimit 0.0 highLimit 1000.0
:addressPV s7:[LOCALSERVER]db1,w6
:addressSP s7:[LOCALSERVER]db1,w26
:sysmbol PLC1Variable4 :PV 360.0 :SP 334.0))))
    )
```

The above two examples are given based on the created ontology and adopt the FIPA-SL content language. It is not necessary to write messages in text form as presented above because it is possible to use *the* JADE *agent content manager* for creating these messages and let the developer creates and manipulates content expressions as Java objects. Ontology is essentially a collection of schemas that typically doesn't evolve during an agent lifetime [16], the JADE

ISBN: 978-1-61208-405-3

agent development platform provides the developer with the required tools and classes to create his application ontology, but this way is being cumbersome with large Ontologies. Fortunately, it is possible to define the ontology using the Protégé tool [19], and then, let the Bean Generator add-on [20] to automatically create the ontology definition class plus the predicates, agent actions and concepts classes. The proposed ontology designed and created by the Protégé Tool. An agent not only interacts with other agents but also it carries out a set of *Internal Behaviors* according to its interaction results with other agents or according to the changes take place in its environment. In the proposed ICN, the agents' internal behaviors can be extracted from the agent responsibility tables and the *use case* diagram. For a control agent the important internal behaviors are:

1. *handleDataChange*: it is a one shot behavior executed periodically to read process data and checks if there is a process data change and if there is, it sends the process changed data to the connected remote operator agents and also sends changed cross variables to other control agents. This behavior invokes another behavior for providing complex higher level control algorithms, which require higher-capability resources that cannot be provided by the basic limited-resources control systems. For instance, these complex computational algorithms can be interpolation, global synchronization and so on.
2. *manageOperatorSetpoints*: it is a JADE (Finite State Machine) FMS behavior implements a defined finite state machine. Figure 6 shows the finite sate machine diagram, which is implemented by this behavior. The behavior is executed just after the control agent receives a setpoint request from a remote agent. This behavior includes four child behaviors each one of them extends the Jade *OneShotBehaviour*. See the implementation phase section. The validity of operator setpoints can be evaluated based on the allowable process variable range (i.e., min and max).

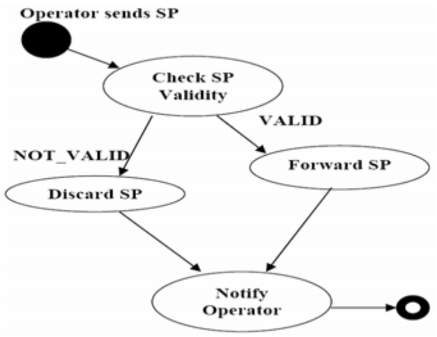


Figure 6. Validation of operator Setpoints.

3. *prepareNewSP:* After a control agent subscribed to other control agents for getting cross reference process data, it continuously receives those cross process data and forward them to this behavior for processing them and calculate new setpoints for specified local process variables. (See the implementation phase section*).*
4. *higherLevelControl:* this behavior is initiated by the handleDataChange behavior if there is any process data change. It is a one shot behavior contains a number of algorithms for processing variables processing. In other words this behavior realizes the dependency relations among control processes variables. For instance, the setpoint of a process variable depends on the actual value of another process variable and the former may be calculated from the later through an interpolation algorithm, which needs higher computing power. Figure 7 presents an illustrative example; Var5.SP is calculated from Var4.PV through an interpolation algorithm executed by the control agent. Many other mathematical complex algorithms can be added to this behavior as required.

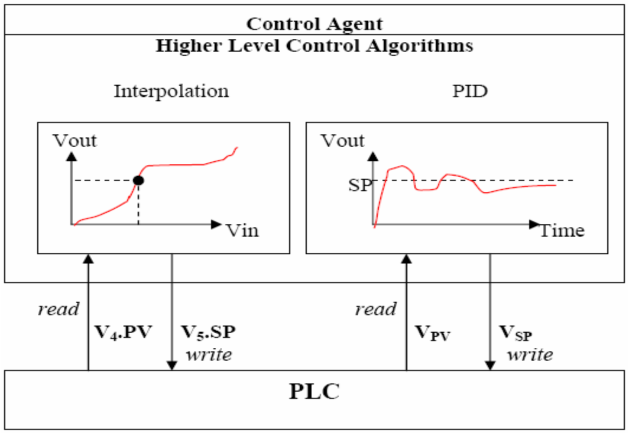


Figure 7. The process variables dependency relations.

This way, control agents provide an extra computational power for the underlying limited resources control systems.

C. *Implementation Phase*

In the implementation phase, all the previously designed constructs and artifacts will be implemented as JADE behaviors. The proposed approach is based on the integration of MAS and OPC protocol, realizing this integration enables us to achieve two goals, first it will be possible to transfer the OPC process data from the process domain to the information domain (MAS). Second, it will be possible to take the benefit of control devices interoperability provided by the OPC process protocol. Using a Java development environment, such as Eclipse [21], frees the developer from caring about modifying related system variables such as CLASPATH and PATH as it does these issues automatically. To connect a Java agent under Eclipse to an OPC server, it is required first to install a MAS platform, such as JADE.

TABLE II. INTERACTION TABLE FOR AN OPERATOR AGENT

Interaction	Resp.	IP	Role	With	When
Search for process service	1	FIPA Request	I	DF	After starting up
Subscribe to a control agent	2	FIPA Request	I	A control agent	After discovering services
Receive Process Actual values	3	FIPA Inform	R	A control agent	always
Receives Process Setpoints	3	FIPA Inform	R	A control agent	always
Receives notifications and Alarms	7	FIPA Inform	R	A control agent	always
Send a Setpoint to a process variable	6	FIPA Request	I	A control Agent	When the operator submit a setpoint through his GUI

TABLE III. INTERACTION TABLE FOR A CONTROL AGENT

Interaction	Resp.	IP	Role	With	When
Register process services	1	FIPA Request	I	DF	After starting up
Subscribe to DF to be notified when control agents register their services.	2	FIPA Subscribe	I	DF	After Starting up
Handle subscriptions from related control agents	3	FIPA Request	R	Control agents	After initializing
Subscribe to related control agents for cross process variables	4	FIPA Request	I	A control agent	After discovering related control agents by DF
Receive cross process data from control agents	5	FIPA Subscribe	I	control Agent	always
handle subscription requests from operator agents for local process data	6	FIPA Request	R	operator Agent	always
receives operator setpoints	8	FIPARequest	R	Operator Agent	When operator send a setpoint

JADE is a software framework fully implemented in Java language, it simplifies the implementation of multi-agent systems through a middleware that claims to comply with the FIPA specifications [22] and through a set of tools that supports the debugging and deployment phase. The agent platform can be distributed across machines with different operating systems and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by creating new agents and moving agents from one machine to another one as/when required. Moreover, JADE is distributed in open source. To run JADE under Eclipse, the developer should add JADE libraries to Eclipse Java build path: (*project→prosperities→Java Build path→Libraries→add external Jars*), then through the Windows file system find *Jade.jar* file in the JADE home. Now Eclipse is ready for creating a new java class that extends *jade.core.Agent* class and start programming the required agent. JADE platform provides to the developers a variety of behavior types. It not only provides support for developing simple behaviors such as *OneShotBehaviour* but also it provides support for developing composite behaviors such *SequentialBehaviour* and *FSMBehaviour*. Furthermore, JADE provides ready to use behaviors for implementing interaction protocols such as request, inform, subscribe, and so on.

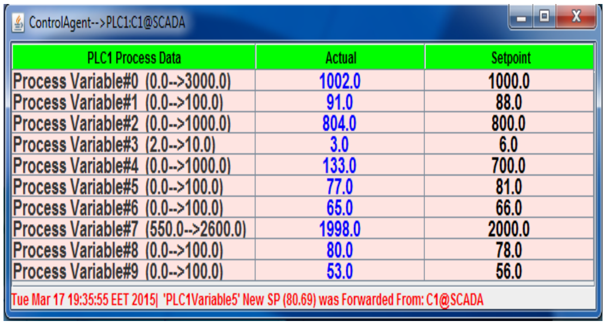
D. *Testing and Evaluation Phase*

The proposed agent-based ICN was tested and evaluated with simulated process OPC data. The OPC server provides a way to access its internal variables without connecting physically to a real control system. Connecting to the OPC server requires the agent to know the *OpcServerHost* and *OpcServerName* settings. In the proposed ICN the later is (OPC.SimaticNet) for Siemens, and the former is (localhost), which means that the OPC server is situated on the same host as the control agent. In other applications, the OPC server can be hosted on a different host on the site LAN; in this case, the control agent will connect to it using DCOM (Distributed COM) [7][31]. Following this behavior, there is another one shot behavior for creating the OPC groups containing the process variables. Each process variable is treated as an OPC item in an OPC group. The address of each OPC item is determined by what is called connection string. For instance, with the used simulation environment, an item address can be like *s7:[@LOCALSERVER]db1,w2*. And for real applications it can be like:

S7:[S7connection1|VFD3\S7ONLINE|02.00,192.168.100.24,02.03,1]db190,w390

The testing and evaluation results after running the system-to-be based on a simulation environment can be summarized as follows:

1. The flexibility and easiness of the proposed development approach can be concluded the adopted ad hoc methodology.
2. Figure 8 presents the control and remote operator agents which provide the required real-time monitoring and supervisory. The designed agent GUI is simple, but can be more complex and user friendly in real applications.


ControlAgent-->PLC1:C1@SCADA

PLC1 Process Data	Actual	Setpoint
Process Variable#0 (0.0-->3000.0)	1002.0	1000.0
Process Variable#1 (0.0-->100.0)	91.0	88.0
Process Variable#2 (0.0-->1000.0)	804.0	800.0
Process Variable#3 (2.0-->10.0)	3.0	6.0
Process Variable#4 (0.0-->1000.0)	133.0	700.0
Process Variable#5 (0.0-->100.0)	77.0	81.0
Process Variable#6 (0.0-->100.0)	65.0	66.0
Process Variable#7 (550.0-->2600.0)	1998.0	2000.0
Process Variable#8 (0.0-->100.0)	80.0	78.0
Process Variable#9 (0.0-->100.0)	53.0	56.0

Tue Mar 17 19:35:55 EET 2015| 'PLC1Variable5' New SP (80.69) was Forwarded From: C1@SCADA

Figure 8. A simple operator GUI designed for each agent in the system.

3. The designed GUI provides the required operator support i.e., real-time process data, alarm service, and trend service. a Figure 9 presents an example process variable trend.

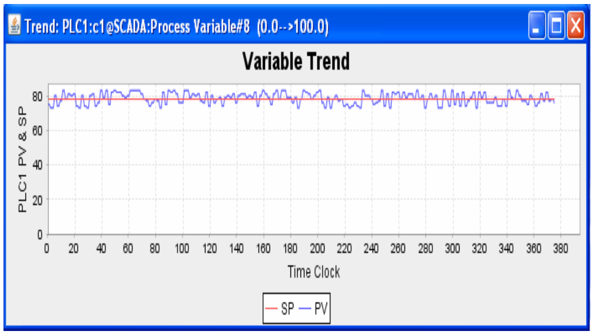


Figure 9. Trend diagram showing a process variable SP and PV.

4. Figure 10 provides an example of the higher level control algorithm embedded inside a control agent. The figure demonstrates how the control agent read a process variable and according to a proper mathematical model (i.e., interpolation, PID algorithm, etc.) the agent continuously calculates the value of another process variable setpoint and send it to the underlying legacy PLC. As shown with the existence of predefined (PLC1Var4.PV, PLC1Var5.SP) points, the value of PLC1Var5.SP can be calculated given the value of PLC1Var4.PV. As shown in the figure, while the value of PLC1Var4.PV increases the value of PLC1Var5.SP decreases.

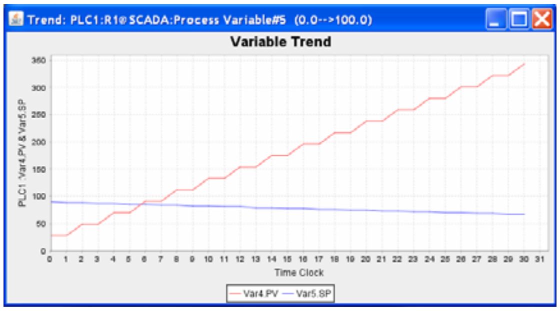


Figure 10. Process variables dependency.

5. Achieving a global synchronization among control processes is another important higher level control activity done by the control agents' cooperation. Figure 11 shows the trend diagrams of three dependent cross process variables, each variable is contained in a different control process but its SP depends on another process variable SP contained on another related control process. The situation shown in Figure 11 demonstrates the automatic tuning of process variables as the SP of the first process variable changes. The first trend in the figure presents the change of the setpoint of a process variable in process (PLC1) and the second and third trends show how other dependent process variables setpoints change accordingly to synchronize the whole production processes.

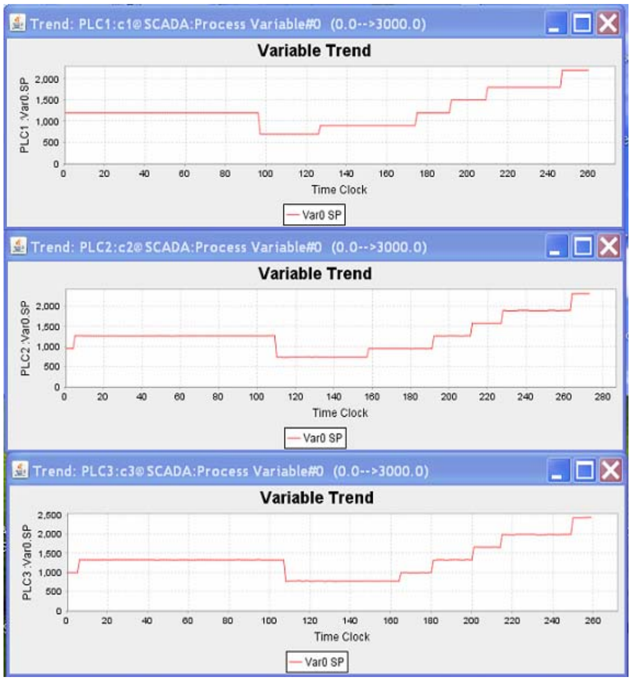


Figure 11. Global synchronization: cross variables dependency.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-405-3

V. Conclusion and Future Work

Agents and multi-agent systems have been applied in many disciplines and they were successful as a new software engineering style for the development of high quality software products. The agent-based applications have a combination of quality attributes, which were difficult to be found in one software application before multi-agent technology. This paper provided a development case study on building a multi-layered agent-based industrial control network, which is an example of highly distributed, open, critical and complex systems. The developed agent-based ICN demonstrates how to realize a distributed control system from logically separated legacy control systems have limited capabilities with lower cost as a main concern. The proposed ICN is a multi-layered industrial network exploits cooperative autonomous agents to supervise and control a distributed control system consists of three processes controlled basically by legacy PLC units. Each PLC unit is assigned to a control agent to provide higher level control algorithms and cooperates with other control agents to achieve a type of global synchronization among control processes and realize the dependency relations among local process variables. Unlike other related work, the proposed ICN is built on a step by step basis from analysis to evaluation to be a comprehensive reference for practical adoption of Agents in the development of ICN. The main conclusion of this research is that the agent-based approach is the promising solution for handling future ICN challenges especially with the evolving topic of the Internet of Things, which concerns devices capable to communicate via the Internet and manipulate an enormous amount of data. As a future work, it is required to apply the agent-based approach to the development of large-scale ICN such as SCADA networks with large number of control agents and remote operator agents.

REFERENCES

[1] H. A. Abbas, Future SCADA challenges and the promising solution: the agent–based SCADA. International Journal of Critical Infrastructures, 10(3), 2014, pp. 307-333.

[2] G. D. M. Serugendo, M. P. Gleizes, and A. Karageorgos, Self-organising systems, 2011, pp. 7-32. Springer Berlin Heidelberg.

[3] G. Weiss, Multiagent systems: a modern approach to distributed artificial intelligence. MIT press, 1999.

[4] M. Wooldridge, An introduction to multiagent systems, John Wiley & Sons., 2009.

[5] E. Oliveira, K. Fischer, and O. Stepankova, Multi-agent systems: which research for which applications. Robotics and Autonomous Systems, 27(1), 1999, pp. 91-106.

[6] D. Weyns, T. Holvoet, and K. Schelfthout, Multiagent systems as software architecture: another perspective on software engineering with multiagent systems. In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems , 2006, pp. 1314-1316, ACM.

[7] E. Diaconescu and C. Spirleanu, Communication solution for industrial control applications with multi-agents using OPC servers, In Applied and Theoretical Electricity (ICATE), 2012 International Conference on, 2012, pp. 1-6.

[8] F. Bellifemine, A. Poggi, and G. Rimassi, “JADE: A FIPA-Compliant Agent Framework.” Proceedings of the Practical Applications of Intelligent Agents and Multi-Agents, April 1999, pp. 97-108.

[9] A. Pereira, N. Rodrigues, and P. Leitão,” Deployment of Multi-agent Systems for Industrial Applications”, 17th IEEE International Conference on Emerging Technologies and Factory Automation, 2012, pp. 1-8.

[10] W. Farai Rupare, L. Nyanga, A. van der Merwe, S. Mhlanga, and S. Matope, “Design Of An Automated Grinding Media Charging System For Ball Mills” , SAIIE25 Proceedings, 9th – 11th of July 2013, ref no. 619, pp. 507-518, Stellenbosch, South Africa.

[11] E. M. Burmakin and B. A. Krassi, "Distributed automation and control systems." International Student Olympiad on Automatic Control 9, 2002, pp. 25-28.

[12] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks", IEEE Communications Surveys &Tutorials, vol 5, no.2, pp. 860-880, Second Quarter 2013.

[13] OPC Foundation, “OPC DA 3.0 Specification [DB/OL]”, Mar.4, 2010

[14] FIPA ACL Specifications [Online]. Available: http://www.fipa.org/repository/index.html, [retreived:3,2015].

[15] M. Nikraz, G. Caire, and P. A. Bahri, A Methodology for the Analysis and Design of Multi-Agent Systems using JADE, May 2006 issue of the International Journal of Computer Systems Science & Engineering special issue on “Software Engineering for Multi-Agent Systems”, 2006, pp. x-y.

[16] OMG UML Specification Version 1.3. Object Management Group, Inc., http://www.rational.com /uml/resources/documentation/index.jtmpl, [retreived:3,2015].

[17] Joseph P. Russell, Java programming for absolute beginner, Prima publishing, 2001, USA.

[18] http://sourceforge.net/projects/jeasyopc/, [retreived:3,2015].

[19] Protégé, http://protege.stanford.edu/, [retreived:3,2015].

[20] Jade Bean Generator add-on, http://protege.cim3.net/cgi-bin/wiki.pl?OntologyBeanGenerator, [retreived:3,2015].

[21] https://www.eclipse.org/, [retreived:3,2015].

[22] http://www.fipa.org/, [retreived:3,2015].

[23] W. Bolton, “Programmable Logic Controllers,” 5th ed., Newnes, 2009 ISBN 978-1-85617-751-1, Chapter 1.

[24] P. D. Anh and T. D. Chau, Component-based Design for SCADA Architecture, International Journal of Control, Automation, and Systems (IJCAS), vol. 8, no. 5, 2010, pp.1141-1147.

[25] H. A. Abbas and A. M. Mohamed, ”Review in the design of web based SCADA systems based on OPC DA protocol”, International journal of computer networks, Malaysia, Vol.2, Issue 6, 2011, pp. 266-277.

[26] A. M. Mohamed and H. A. Abbas, “Efficient Web Based Monitoring and Control System”, Proceedings of the Seventh International Conference on Autonomic and Autonomous Systems, ICAS 2011, May 22-27, 2011, pp. 18-23, Venice, Italy.

[27] M. Metzger, and G. Polakow. "A survey on applications of agent technology in industrial process control." Industrial Informatics, IEEE Transactions on 7.4, 2011, pp. 570-581.

[28] F. Bergenti and E. Vargiu. "Multi-Agent Systems in the Industry. Three Notable Cases in Italy." WOA. 2010.

[29] A. F. Sayda, Multi-agent systems for industrial applications: design, development, and challenges. INTECH Open Access Publisher, 2011.

[30] M. Pěchouček and V. Mařík. "Industrial deployment of multi-agent technologies: review and selected case studies." Autonomous Agents and Multi-Agent Systems 17.3 (2008): 397-431.

[31] R. Kondor. (2007). OPC and DCOM: Things you need to know. Available: http://xlreporter.net/download/OPC_and_DCOM.pdf

Application Design and Profiling of Stream Processing

Veronica Gil-Costa, Jair Lobos
Universidad Nacional de San Luis, CONICET
San Luis, Argentina
Email: {jlobos,gvcosta}@unsl.edu.ar

Mauricio Marin
DIINF, University of Santiago, Chile
Center for Biotechnology and Bioengineering, Chile
Email: mauricio.marin@usach.cl

***Abstract*—Stream processing has recently caught the attention of many researchers and engineers, mainly because of the continuous growth of information generated by users. Stream processing platforms allow processing and analyzing real time data, which helps to make decisions faster. In this work, we determine the most relevant tasks performed by the distributed stream processing platform named S4. To this end, we develop a pool of benchmark applications and we make a profiling of their execution using the S4 platform. Results show that the most relevant operations are related to the control and manipulation of threads.**

Keywords—stream processing; S4; profiling.

I. INTRODUCTION

The world has become fully connected. There is a large number and variety of data resources available from hardware and/or software systems. There are numerous industries where everyday processes and interactions with customers generate millions of events that produce traces, with information regarding user’s activity. These traces contain valuable information for understanding and optimizing processes. In addition, those traces can be used to detect anomalies, to predict the behavior and trends of customers, among other activities that can improve the productivity of a company or institution.

The events are collected from users actions form a continuous amount of data stream. Some examples can be found in: market analysis; telecom call detail records; video surveillance systems; vital signs of a patient in a medical system; intrusion records system networks; the behavior in a system of Web 2.0, among others. In all these applications it is necessary to collect, process and analyze the data stream, and then generate results or produce some specific actions. An important feature of these applications is that the analysis must be done in real time.

There are many stream processing platforms such as SPC (Stream Processing Core) [2], Storm [3], Esc [4] and D-Stream [5]. Some research works like TimeStream [13], StreamCloud [14][15] and CEC [16], have endeavored to present solutions to the problems of load balancing and fault tolerance of the stream processing process.

Recently, a general-purpose distributed platform designed to analyze massive data processing called S4 (Simple Scalable Streaming System) was proposed by L. Neumeyer, et. al. [1]. The S4 world-view is that streams are passed through a graph (DAG) formed by processing elements (PEs), which are connected each other in a downstream manner. Each PE performs a given primitive operation on the received stream and generates output streams. Data is routed through the PEs by means of keys, which are specified by users.

In this work, we develop and test a set of applications covering different computation/communication aspects using the S4 platform. We aim to understand the flow of events processed in those applications with different data streams, to determine which are the most repetitive and costly tasks executed by the S4. We obtain relevant metrics by developing prototypes for each application, which are used as benchmark to detect bottlenecks in both communication and computation operations. The performance information obtained in this work can be used to propose improvements to the stream processing platform itself. By determining the relevant operations executed by the S4 platform and their costs, it is possible then to introduce these costs into a simulation model as the ones presented in [6] to design and test new algorithms without affecting the actual platform running in production. To this end, we developed a pool of benchmark applications built with different characteristics including processing and communication complexities in order to determine the most relevant operations.

Additionally, the results obtained though the benchmark applications can be used in elastic stream processing programming environments [4], where developers can detect possible bottlenecks of PEs, make decisions and take action in advance. In this case, the knowledge obtained by executing the pool of benchmark applications, can aid to determine which PEs should be replicated.

This paper is structured as follows. Section 2 describes stream processing and the S4 platform. Section 3 briefly describes profiling and the tool used in this work. Section 4 describes the pool of benchmark applications used to detect the most relevant operations. Section 6 shows the results. Finally, Section 7 presents the conclusions and future work.

II. STREAM PROCESSING

In this section we discuss the main properties of stream processing, when stream processing makes sense, and how it fits into big data architectures. We also describe the S4 stream processing platform, used to test the benchmark applications presented in this paper.

A. Streaming Processing and Big Data

Big data is commonly defined as the three Vs: Volume, Velocity and Variety [17]. It is used to describe the exponential growth and availability of structured and unstructured data. A more recent, definition states that "Big Data represents the

Information assets characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value" [18].

On the other hand, stream processing is used for fast data requirements, which includes tacking the velocity of processing a huge variety of data in real time. Therefore, both big data and stream processing can complement each other.

Stream processing was first used for finance problems. Today, it is used in almost all industries where stream data is generated by human activities or automatically by sensors. Events are generated on-line in unpredictable time instants. The union of events forms a continuous stream of information that may have dynamic variations in intensity of traffic. In this context, the process used to store and organize/index events in a conveniently way to then process them in batch can be very costly given the huge volume of data and the amount of computational resources required for processing them. But even if this is feasible, it is often desirable or even imperative to process the events as soon as they are detected to deliver results in real time.

In particular, stream processing corresponds to a distributed computing paradigm that supports the process of gathering and analyzing large volumes of heterogeneous data streams to assist decision making in real time.

Stream processing appears as result of the rigorous data management, which is increasingly demanding because of the information generated by business and scientific applications, which are fully linked to the technological progress. It is also related to the advance in hardware and software databases; the management of large amount of data in distributed systems; the use of techniques such as signal processing, statistics, data mining and optimization theory.

Stream processing aims to process data in real time and in a fully integrated way, to provide information and outcomes for consumers and/or end users. Also, it aims to integrate new information to support decision making in the medium and long term.

The high volume of event flows coming from different data sources makes it impossible to store this information, such as model-based on data warehouse where all the data is stored and then to make the appropriate processing and analysis.

Stream processing applications requires fulfilling certain performance requirements in terms of latency and throughput. Specifically, processing must keep up with the rate of incoming data, while it provides a high level of quality of analysis of results as fast as possible. Additionally, the application components and infrastructure must be fault-tolerant.

B. *S4 - Simple Scalable Streaming System*

S4 acronym for "Simple Scalable Streaming System" is a system of general purpose, distributed, scalable, which allows applications to process data flows continuously without restrictions [1]. S4 is inspired by MapReduce [7], designed in the context of data mining and machine learning algorithms of Yahoo! Labs for on-line advertising systems.

In S4, each event is described as a pair (key, attribute). PEs are the basic units and messages are exchanged between them. The PEs can send messages or post results. PEs are allocated in the so-called processing nodes (PNs) servers. The PNs are responsible for: a) receiving incoming events, b) routing the events to the corresponding PEs and c) dispatching events through the communication layer. The events are distributed using a hash function over the key of the events. Furthermore, the communication layer uses Zookeeper [8], which provides management and automatic replacement clusters if a node fails.

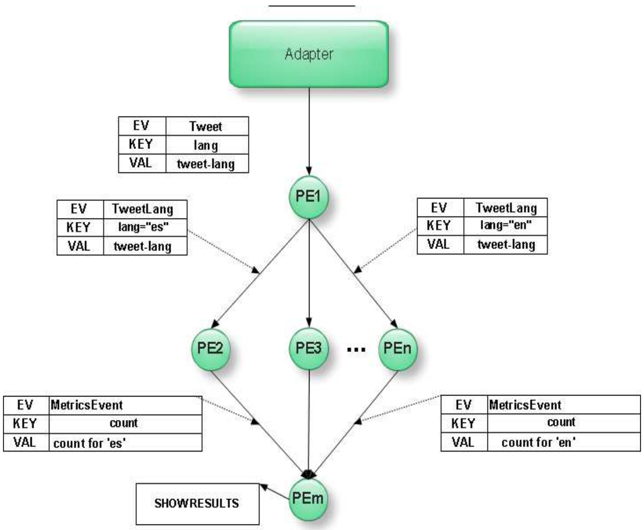


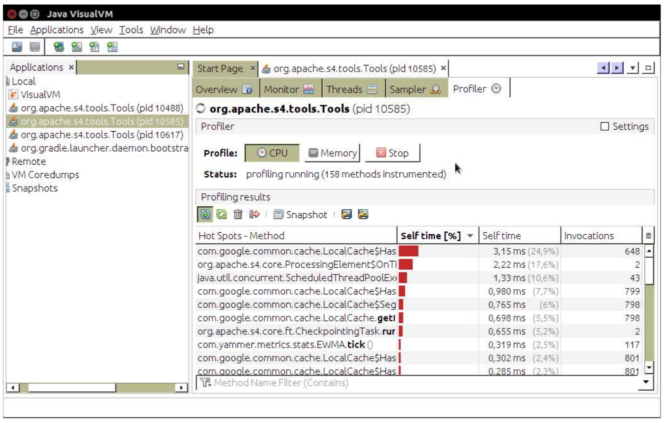
Figure 1. S4 application design (lang-count example).

To run an application with S4, we need to deploy an Adapter application. Adapters are S4 applications that can convert external stream into stream of S4 events. Figure 1 shows a simple Tweet language count for Twitter. In this example, input events contain a language descriptor for a tweet from Twitter. The Adapter gathers tweets from twitter and filters only the language descriptor. Then, the Adapter sends an event to PE1. PE1 listens for Tweet events with all possible keys. For each possible key, PE1 emits a new event of type TweetLang. PE2-n listen for TweetLang events emitted with the key lang. For example, PE1 emits an event with key lang=”es”. PE2 receives all events of type TweetLang keyed lang=”es”. If the PE corresponding to the emitted key exists, the PE is called and the counter of language is incremented. Otherwise, a new PE is instantiated and linked to the new key. Whenever a PE increments its counter, it sends the update count to the PE called PEm and this show the results.

III. Profiling And Tools

A profiler generates the division of the logical structure of the applications so that user can understand how a particular run of the application is performed using relevant information regarding execution time and memory usage.

Using a system profiler we can obtain a model to predict scaling factors as characteristic functions of the applications and hardware parameters [9]. Currently, there are several tools available to perform system profiles. S4 requires a JAVA profiler, among which we can highlight: Profiler4j [10] jvisualvm [11] and Java Profiler Tool [12]. By using these tools and the applications described in the next section, we intend to obtain a S4 profile to determine which are the most costly and the most relevant operations executed by the S4 stream processing platform.


Java VisualVM
File Applications View Tools Window Help
Applications ×
Local
VisualVM
org.apache.s4.tools.Tools (pid 10488)
org.apache.s4.tools.Tools (pid 10585)
org.apache.s4.tools.Tools (pid 10617)
org.gradle.launcher.daemon.bootstra
Remote
VM Coredumps
Snapshots
Start Page ×
org.apache.s4.tools.Tools (pid 10585) ×
Overview
Monitor
Threads
Sampler
Profiler
Profiler
□ Settings
CPU
Memory
Stop
Profiling results
Snapshot
Method Name Filter (Contains)

org.apache.s4.tools.Tools (pid 10585)

Profile: CPU

Status: profiling running (158 methods instrumented)

Hot Spots - Method	Self time [%]	Self time	Invocations
com.google.common.cache.LocalCache$Has		3,15 ms (24,9%)	648
org.apache.s4.core.ProcessingElement$OnTi		2,22 ms (17,6%)	2
java.util.concurrent.ScheduledThreadPoolEx		1,33 ms (10,6%)	43
com.google.common.cache.LocalCache$Has		0,980 ms (7,7%)	799
com.google.common.cache.LocalCache$Seg		0,765 ms (6%)	798
com.google.common.cache.LocalCache.**getl**		0,698 ms (5,5%)	798
org.apache.s4.core.ft.CheckpointingTask.**rur**		0,655 ms (5,2%)	2
com.yammer.metrics.stats.EWMA.**tick** ()		0,319 ms (2,5%)	117
com.google.common.cache.LocalCache$Has		0,302 ms (2,4%)	801
com.google.common.cache.LocalCache$Has		0,285 ms (2,3%)	801

Figure 2. Snapshop of the jvisualvm tool.

Figure 2 shows the environment of the jvisualvm tool when measuring the CPU utilization. The CPU Profiling is used to test the performance of the application and it gives detailed information about the total execution time and the number of calls for each method. In the same way, the Memory Profiling is used to analyze the memory usage, by showing the total number of executed methods or objects and the amount of bytes assigned to each one.

IV. BENCHMARKS APLICATIONS

This section describes a pool of applications developed to run on the S4 platform to perform tests in order to obtain a system behavior profile. These benchmark applications will help us to determine the most costly and most relevant operations. Each application has different levels of complexity on the tasks performed and different levels of communications. The applications described below can be classified into the following categories: 1) High communication, 2) dynamic creation of processing elements, 3) high, medium and low computation. Though this classification, we obtain the S4 operation costs for each type of benchmark application.

A. *Ping-Pong*

Figure 2 shows a basic communication structure between two processing elements. PE A, named “sender”, generates a new message and sent it to PE B, named “receiver”. Finally B replies this message to A. This benchmark program uses different messages sizes. Each event uses messages between 8 and 256 characters size. This application is classified as low computation but with high communication between the processing elements.

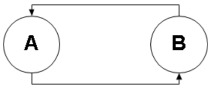


Figure 3. Ping-Pong application.

B. *Router*

The next application is called “Router”. This application generates random values in the Adapter module. The Adapter sends messages to the PE R, which determines if this value is an even or an odd number. If the received number is even, R sends a message (an event) to the Processing Element named Even. Otherwise, if the number is odd, R sends the event to the Processing Element named Odd. Both Odd and Even PEs make a count of the received elements. When all messages are dispatched, each PE sends its results to the Processing Element named Res, and this PE shows the final results. Figure 3 shows the flow of events and the PEs of this application. This application is classified as low computation but with high communication between the Adapter and the PEs.

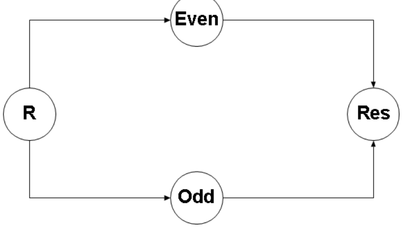


Figure 4. Router application.

C. *Counter of Tweets and Re-Tweets*

This application works with the Twitter API to get tweets from “Twitter’s global stream of Tweet data”. This application is connected to the data repository to extract tweets, which are processed by the Adapter. The Adapter receives the tweets, creates an event and sends them to the PE named T. This last PE classifies the tweet as “No-Re-Tweet” when it corresponds to message that has been posted for the first time, or “Re-Tweet” when it corresponds to a message that has been re-posted by other users.

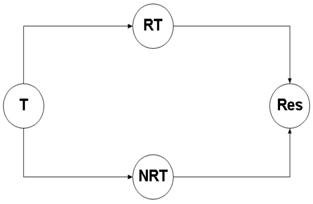


Figure 5. Counter of Tweets and Re-Tweets application.

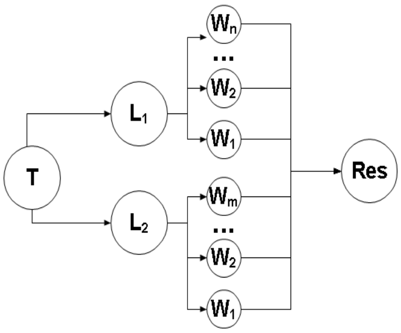


Figure 6. Language word counter application.

Once the classification process is finished, the message is sent to the corresponding PE (RT or NRT), which extracts the list of hashtag (represents an idea, it is considered as metadata), and stores the five most frequent hashtags. This information is sent to the Processing Element named Res. The Res PE summarizes the results received. Figure 4 shows the corresponding diagram for this benchmark application. This application is classified as *high communication* and as *medium computation*.

ISBN: 978-1-61208-405-3

D. *Language word counter*

This application works with Twitter tweets. The differences with the previous application are that the event classification process is done by tweet language instead of No-Re-Tweet or Re-Tweet classification; and this application counts the number of words that has a tweet. With this process we stress the system because an additional PE is generated for each new word entered into the system. Namely, dynamic PEs are created every time a new word is found inside a tweet. Figure 5 shows the diagram for this benchmark application. The Processing Element named T, classifies the events into two groups L1 and L2, corresponding to languages Spanish and English. Each Processing Element used for language classification gets tweet and splits them into words. Each word is sent to its corresponding Processing Element W. If there is no PE for the received word a new PE is created. The words are counted in each PE and the results are sent to the Res PE. This application has a high communication cost and has dynamic creation of processing elements.

E. *Clasifficator for people's needs in a post-disaster scenario*

The classification process is composed of 4 steps: Recollecting, Filter, Relevance and Ranking. The recollecting step focuses on collecting data from the source to retrieve data from the Twitter API. The filter operator exploits a Naive Bayesian model to identify if tweets are objective or subjective. To create these models an automatic classification is performed through bags of positive and negative words. The bag of words were manually created by developers and validated by undergraduate students based on the information of other disaster tweets datasets. Objective tweets have a higher value, because they are more reliable than the subjective ones. If the tweet is subjective it is checked whether it is positive or negative in order to benefit the tweets based on the identified characteristics by applying weights constants in the ranking process.

The topic step is used to identify whether the information is coming from a trustworthy source or not. Trustworthiness is calculated in two dimensions: author information and tweet information. From the author side, information such as the number of tweets generated, the number of followers/followees and an account verification state are considered to calculate the reputation of the author. From the tweet side, the number of re-tweets, favorite marks, and the associated timestamp are exploited to calculate its reputation.

During the ranking step a normalization process of the obtained values is performed. This was computed every certain number of tweets, to get statistics such as the maximum number of followers, the number of favorites, etc. This data is used to normalize each tweet.

F. *DownStream Web Search Engine*

Typically, web search engines are composed by three services devised to quickly process user queries in an on-line manner: Front-Service (FS), Caching-Service (CS) and Index-Server (IS). In such systems a query submitted by a user goes through different stages. Initially, it is received by the FS, which redirects the query to the CS. The CS checks whether the same request has already been performed and verifies if the result (document IDs) are stored in the cache memory of the server. The CS can answer to the FS with a cache hit. In this case, the CS sends the query results to the FS, which builds the Web page with the query results and sends it to the user. Otherwise, if the CS sends a cache-miss to the FS, the FS re-routes the query to the IS, which will compute the top-k document results.

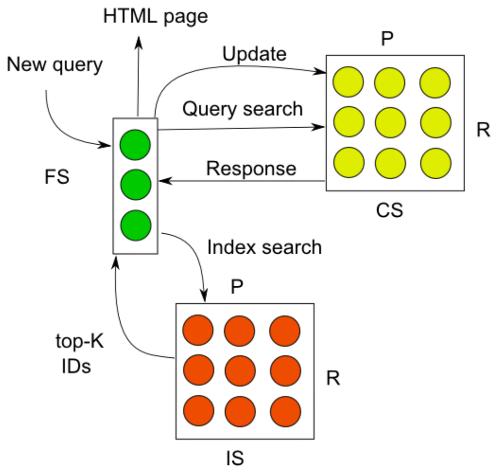


Figure 7. Components of a web search engine.

These services are deployed on a large set of processors forming a cluster of computers. They are implemented as arrays of P × R processors, where P indicates the level of data partitioning and R the level of replication of data. Hence, this architecture makes a high usage of partitioning and redundancy to enhance the query response time and throughput. For instance, each query is assigned a unique partition of the CS using a hash function, and different CS nodes can be associated with a partition to answer a query (see Figure 6).

Figure 7 shows a web search engine application designed for the S4 platform. The diagram represents the query flow through the web search engine components. Each component is composed by a set of different PEs. This application has the following structure: the FS is divided into three sub-services FS1, FS2 and FS3. Each group executes the tasks performed by the FS in different moments of the query processing process. In other words, the group named FS1, executes the tasks required to route an incoming query to the CS. The group named FS2, executes the tasks required to route the query to the IS, if no cache hit was reported, or the tasks required to build the query answer and send it back to the user, otherwise. The CS is divided into two groups CS1 and CS2. The first group detects cache hits and the second group updates the cache with query results. There is only one group of IS, because this service is used only once during the query process, to compute the top-k document results for queries. This application is classified as high in communication and high in computation.

V. RESULTS

In the following, we show results obtained by executing all benchmark applications described in the previous section. During each execution, we detect the most costly and relevant operations. The profiling execution was captured by the jvisualvm tool [11]. Results were obtained in a cluster of 16

64-bits CPUs Intel Q9550 Quad Core 2.83 GHZ and 4GB DDR3 RAM 1333 Mhz. Additionally, to verify the results, tests were also performed on an Intel I5-4200U 1.6 GHZ and 8GB DDR3 RAM 1600 Mhz.

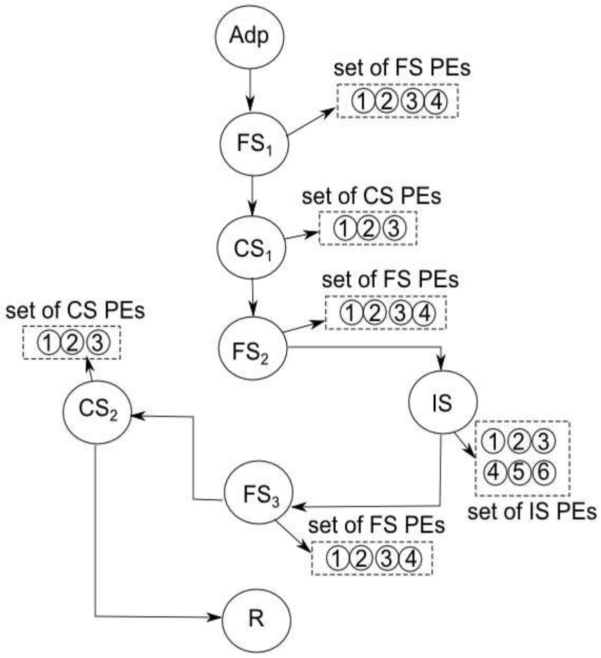


Figure 8. Web search engine diagram for the S4 platform.

TABLE I. COMMUNICATION OPERATIONS

Package	**Operations**		
	Class	***Method***	***Cost(ms)***
comm.staging	BlockingThread PoolExecutor Service	RunneableWithPermit Release.run	0.0019
		execute	0.0072625
comm.tcp	TCPEmitter	Init	--
		getPartitionCount	0.0005988
comm.tcp	TCPListener	EventDecoderHanler.messageReceived	0.0036462
comm.topology	**ZKRemote Stream**	**createStreamPaths**	**38.6**
		getPath	--
		getCollectionName	--
		addInputStream	**93.3**
		update	--
		refreshStreams	--
comm.topology	ZkClient	readData	--
		getChildren	--
comm.topology	ZNRecord	getSimpleField	--
		putSimpleField	--
comm.topology	PhysicalCluster	getNodes	--
comm.topology	Stream Consumer	Equals	--
		hashCode	--
s4.comm	DefaultHasher	hash	0.0002567

For the sake of simplicity, we present the average results obtained for each S4 operation using all benchmark application described in section IV. We cover all the S4 tasks.

A. Communication Operations

In this section, we detect and evaluate the most commonly and/or costly operations used for communication. Note that although S4 communication classes use objects and methods from other libraries or packages, we focus only on those belonging to the S4 platform.

Table I shows the S4 communication operations used to create nodes and to obtain information about clusters. TCP communications are started, creating path for the streams, adding streams and using a hash function to determine the route of events. Communication operations not relevant because of their low costs, do not have time costs in the fourth column of Table I. The results presented are average times obtained with all benchmark applications.

Results show that the most expensive communication operations are **addInputStream,** which publishes interest in a stream, by a given cluster and **createStreamPaths** which creates a zookeeper node to produce and consume streams.

TABLE II. COMPUTATION OPERATIONS

Package	**Operations**		
	Class	***Method***	***Cost(ms)***
Core	**S4Boostrap**	**run**	**1762**
Core	App	init	412
		start	32.4
		createInputStream	94.6
Core	ProcessingElement	handleInputEvent	0.0866573
		isCheckpointable	0.0008995
		recover	0.026
		getInstanceForKey	0.0647487
		setApp	--
		setName	--
core	Stream	StreamEventProcessingTask.run	0.1644673
		put	0.0200624
core	DefaultCoreModule	loadProperties	--
		configure	17.5
		provideTmpDir	--
core	ReceiverImpl	checkAndSendIf NotLocal	0.0034892
		receive	--
base	Key	addStream	--
		get	0.0076047

B. Computation Operations

The most important S4 computation operations are related to the initialization of objects, creation of communication objects for the communication layer, and related to control and manipulation of PEs and events arriving to the PEs. Table II shows the most relevant transactions.

Table II, shows that the Bootstrap is the most expensive process in terms of time consuming. This operation loads the application into main memory and starts it execution. Hence, it takes a larger time compared to others operations. The application Initialization takes a couple of milliseconds. The methods related with the processing elements creation do not take a significant execution time.

Table II, shows that the Bootstrap is the most expensive process in terms of time consuming. This operation loads the application into main memory and starts it execution. Hence, it takes a larger time compared to others operations. The application Initialization takes a couple of milliseconds. The

methods related with the processing elements creation do not take a significant execution time.

TABLE III. S4 MOST USES METHODS (STATIC VS DYNAMIC)

Operation	***Static***	***Dynamic***	***Diff***
Receive message	0,00029299	0,00087835	-199,78%
Create PE	0,02425	0,0182	24,94%
Set Key	1,518	0,021	98,61%

TABLE IV. EXTERNAL TASKS

Package	**Operations**		
	Class	***Method***	***Cost(ms)***
java.net	SocketInputStream	read	83.901729
java.net	SocketOutputStream	socketWrite	0.244189128
java.util.concurrent	**ThreadPoolExecutor**	**getTask**	**2213.0740740**
java.io	ObjectStreamClass	lookup	0.0001040
java.io	BufferedOutputStream	write	0.0000272
sun.nio.ch	EPollArrayWrapper	poll	160.5876465
java.util.concurrent.locks	LockSupport	parkNanos	665.0635514
		park	47.1500057

The operations concerning stream process have no big impact on the S4 platform. Although, these operations do not take much time, they must take into consideration the method that allows connection to Zookeeper for communications, as well as the method that checks whether the cluster, which will be used for communication, is local or not.

Table III shows results for the operations concerning to the most used methods between static and dynamic PEs creation. The operation of receiving a message is most expensive for dynamic PE creations applications than statics applications. However creating a PE and setting a key for each processing elements is most expensive for statics applications.

C. External Tasks

Table IV shows the external tasks used by the S4 platform. These methods are basically used for the manipulation, use of threads and invocations to methods of additional packages. Table IV shows the most relevant transactions obtained by running the benchmark applications. These methods are basically Java core packages (java.net, java.util and java.io) for manipulating stream. An external method to highlight is EpollArrayWrapper, which manipulates a native array of epoll event. Another important method is LockSupport, used for thread blocking with locks and synchronizations. The most costly operations is the getTask from ThreadPoolExecutor, which controls the blocking of tasks of the threads.

VI. CONCLUDING REMARKS AND FUTURE WORK

We presented a profiling study for the S4 Stream processing platform to determine the most costly and relevant operations. To the best of our knowledge, this is the first work concerning benchmark for streaming processing. We developed a pool of applications with different complexity. We used jvisualvm to make the profiling of the executions. Results show that the most relevant operations executed by the S4 platform are related to the creation of applications and the manipulation of events. The most costly operations are Thread control and Thread manipulation.

Future work includes the design and implementation of a simulator for the S4 platform, whose parameters will be set by the results obtained in this work.

ACKNOWLEDGMENT

This work was funded in part by Conicyt-Mincyt grant CH1204, Partially supported by CeBiB basal funds FB0001, Conicyt, Chile, and STIC-AMSUD RESPOND.

REFERENCES

[1] B. Robbins, A. Nair, and A. Kesari, “S4: Distributed Stream Computing Platform. Leonardo Neumeyer”, in ICDM 2010 pp. 170-177.

[2] L. Amini, H. Andrade, R. Bhagwan, F. Eskesen, R. King, P. Selo, Y. Park, and C. Venkatramani, “Spc: a distributed, scalable platform for data mining”, in DMSS&P, 2006, pp. 27–37.

[3] Storm. [Online]. Available: https://github.com/nathanmarz/storm/wiki, retrieved: March, 2015

[4] B. Satzger, W. Hummer, P. Leitner, and S. Dustdar, “ESC: Towards an Elastic Stream Computing Platform for the Cloud”, in CC, 2011, pp. 348–355.

[5] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, “Discretized streams: fault-tolerant streaming computation at scale”, in SOSP, 2013, pp. 423–438.

[6] V. Gil-Costa, J. Lobos, R. Solar, and M. Marin, “AMEDS-Tool: An Automatic Tool to Model and Simulate Large Scale Systems”, in Summer Simulation Multi-Conference, 2014, pp 20:1--20:8.

[7] H. Andrade, B. Gedik, and D. Turaga, “Fundamentals of Stream Processing Aplications Design, System and Analytics”, Cambridge University Press, 2014.

[8] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, “Zookeeper: wait-free coordination for internet-scale systems”, in USENIXATC, 2010, pp 11–11.

[9] S. Graham, P. B. Kessler, and M. K. Mckusick, “Gprof: A call graph execution profiler”, in CC, 1982, pp.120-126.

[10] Profiler4j: http://profiler4j.sourceforge.net, retrieved: March, 2015

[11] Java Virtual Machine Monitoring, Troubleshooting, and Profiling Tool. http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/, retrieved: March, 2015

[12] Java Profiler Tool. http://www.semanticdesigns.com/Products/Profilers/JavaProfiler.html, retrieved: March, 2015

[13] Z. Qian, Y. He, C. Su, Z. Wu, H. Zhu, T. Zhang, L. Zhou, Y. Yu, and Z. Zhang, “Timestream: reliable stream computation in the cloud”, in EuroSys, 2013, pp. 1–14.

[14] V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, and P. Valduriez, “Streamcloud: An elastic and scalable data streaming system”, in Trans. on PDS, vol. 23, no. 12, 2012, pp. 2351–2365.

[15] R. C. Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch, “Integrating scale out and fault tolerance in stream processing using operator state management”, in SIGMOD, 2013, pp. 725–736.

[16] Z. Sebepou and K. Magoutis, “Cec: Continuous eventual checkpointing for data stream processing operators”, in DSN, 2011, pp. 145-156.

[17] D. Laney, “3D Data Management: Controlling Data Volume, Velocity and Variety”, Gartner, 2001, pp. 1-4.

[18] A. De Mauro, M. Greco, and M. Grimaldi. “What is big data? A consensual definition and a review of key research topics”, in AIP, 2015 pp. 97–104.

Human Friendly Autonomous Robot using Dempster-Shafer Sensor Fusion and Velocity Potential Field Control

Dan-Sorin Necsulescu, Yu Hu,
Department of Mechanical Engineering,
University of Ottawa,
Ottawa, Canada
e-mail: necsu@uottawa.ca, yhu061@uottawa.ca

Jurek Sasiadek
Department of Mechanical and Aerospace Engineering,
Carleton University,
Ottawa, Canada,
e-mail: jurek_sasiadek@carleton.ca

***Abstract*— In this paper, a human friendly autonomous robot, was presented. Navigation of this robot applies sensor fusion technique based on Dempster-Shafer method and velocity potential field. The controller of the autonomous mobile robot is designed to lead the robot toward the goal while avoiding collisions in complex environments and maintain human safety as a first priority. The approach is based on Dempster-Shafer sensor fusion of signals from sonar and passive infrared sensors to allow the robot to identify human presence. A velocity potential field robot controller is formulated for the case of avoiding collisions while giving higher priority to collision avoidance with humans as opposed of objects. Simulations and experiments illustrate the performance of the approach in extreme situations.**

Keywords- human friendly robots; Dempster-Shafer evidence theory; sensor fusion; human safety.

I. INTRODUCTION

Recently, with rise of the cost of labor, robots play more important roles than before. To become more widely applied in the real world, it is necessary that the mobile robots have much more artificial intelligence to work in more complex environments, which might include humans or vehicles with unknown and unpredictable motions, such as factory or home environments. To achieve this goal, the robot has to be able to identify and distinguish the obstacles, and the human and avoid collisions with both of them. Furthermore, the robot should always put human safety as its first consideration.

In this paper, a sensor fusion approach based on Dempster-Shafer evidence theory is combined with velocity potential field approach used for robot control. The justification of this choice is the ability of Dempster-Shafer evidence theory to include supporting evidence, refuting evidence and an uncertainty interval, that permit a suitable use of expert knowledge regarding autonomous robots navigation issues. The detection area of the robot is subdivided into several zones. The sensor fusion approach is employed to fuse both uncertain observations of sonar sensors and passive infrared sensors and to estimate the probability of being occupied by a human in each zone [1] [2]. Then, a human friendly mobile robot navigation approach is used for robot motion control. The controller is based on the velocity potential field method (VPF), which is used to lead the robot moving to the goal avoiding the obstacles [3]. For the novel controller, in this case an improvement of the VPF allows the robot to avoid the human a first priority and only afterwards the obstacles. This novel VPF approach was not applied before for robot collision avoidance. Dempster-Shafer evidence theory permits to fuse outputs from various sensors and then provide the VPF controller with the required distances between the robot, the obstacles and the goal. The paper focuses on the novel results presented in the thesis [14].

The paper presents in Section 2 the Demster Shafer evidence theory used for sensor fusion. Section 3 focuses on the model used for the calculation of the avoidance distance to obstacles. Section 4 presents velocity potential field approach for robot navigation in the presence of humans. Simulation results are the topic of section 5, while experimental results are presented in Section 6, followed by conclusions in Section 7.

II. SENSOR FUSION METHOD BASED ON DEMPSTER-SHAFER EVIDENCE THEORY

In this paper, two types of sensors are used to help the robot sense the environment. The first one is the Passive Infrared Sensor (PIR sensor), which senses the heat emitted by humans. PIR sensor is, however, not sufficiently accurate. Performing hundreds of experiments, in at about 10 % of the experiments the human sensor did not work well. Besides, PIR sensor might identify warm air generated by a heat source as coming from a human. The second type of sensor is the ultrasonic sensor. It emits high frequency sound waves to the objects and then receives them to determine how far they are. Since human is the first priority of the robot, information which is collected by different kinds of sensors is combined to identify human with a higher probability [4]-[9].

Dempster-Shafer evidential theory (D-S theory) was chosen to support the probability calculation given its ability when the sensors contributing information cannot associate a 100 % probability to their output decisions. The algorithm captures and combines whatever certainty exists in the object-discrimination capability of the sensors. Knowledge from multiple sensors about events (called propositions) is combined using D-S theory to find the intersection or conjunction of the propositions and their associated probabilities [4], using

$$m(C_k)=\frac{\sum_{A_i\cap B_j=C_k;C_k\neq 0} m(A_i)m(B_j)}{1-\sum_{A_i\cap B_j=0} m(A_i)m(B_j)} \qquad (1)$$

where A_i and B_j are the focal elements of m_A and m_B, respectively. m_A and m_B are the Basic Probabilities Assignments (BPA) which are combined while C_k are the focal elements of the combined BPA [12].

The frame of discernments is in this case a set of cells in the occupancy grids as shown in Figure 1. The eight grids are able to cover the human sensors detection area.

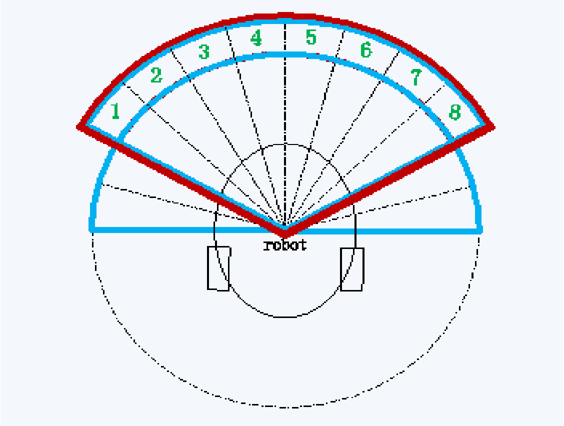


Figure 1. Human Sensors Detection Area

Both ultrasonic sensors and PIR sensors keep watching and each sensor contributes by assigning its BPA over its own frame of discernments [10] [11].

The D-S theory is used to combine two focal elements BPAs.

Sensor fusion is an iterative process with a time step chosen of 0.05s, described as follows:

1. Read the ultrasonic sensor outputs looking for the presence of a human. (assign the BPAs to each grid)
2. Read the human sensors outputs.(assign the BPAs to each grid)
3. Combine the final results of both kinds of sensors using D-S theory after one time step.
4. Calculate the probability (the combined BPAs) that quantifies how probable the grid is to be occupied by a human.
5. Compare the probability value with the setting value and make the decision whether there is a human in the grid and then generate a corresponding response.
6. Initialize the BPA of the occupied grid to zero to restart the all progress.
7. Repeat from step one again until the mission is completed.

III. CALCULATION OF THE AVOIDING DISTANCE

The robot, the human zone and obstacle zone, shown in Figure 2, were built around the robot to avoid collision even in the worst case when the robot has the least amount of time to avoid a head-on collision. We denote the human zone from Figure 2 as A_0 the obstacle zone as A_1 and the work area excluding A_0 and A_1 as A. The shapes of the zones were designed based on the positions of the sensors.

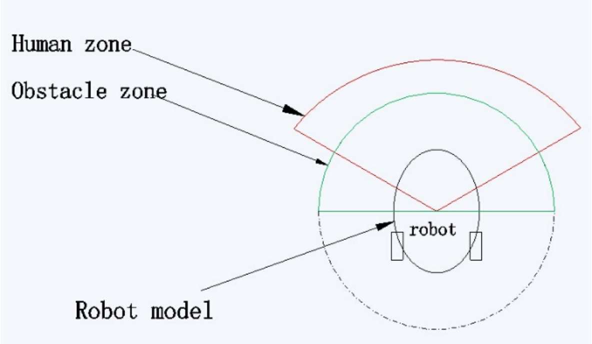


Figure 2. The Human Zone and the Obstacle Zone

The human zone was designed for human avoidance. In the worst human avoidance case, the human and the robot were moving toward each other.

Once the robot detected the human in its human zone, it turns into a different direction, from the original direction toward the goal, to avoid the collision. Given the human size and velocity of the human and the robot

$$r_h = 0.4m \qquad (2)$$

the calculation is based on

$$\bar{\delta}_{mry}=\bar{V}_{mr}\times(\Delta t_1-t_a)+\int\int_{t_a} a_{mry}dt\geq r+\rho_h \qquad (3)$$

$$\bar{V}_{mr}\times(\Delta t_1-t_a)+\frac{1}{2}\bar{a}_{mry}t_a^{\ 2}=\delta_{mry}\geq r+\rho_h \qquad (4)$$

where is the maximum velocity of the robot, t_a is the time that robot needs to accelerate from zero velocity to $\bar{V}_{mr}$ in Y-direction, $\bar{\delta}_{mry}$ is the distance that the robot is moving in Y-direction, r is the radius of the robot, a_{mry} is the robot

ISBN: 978-1-61208-405-3

acceleration magnitude in Y-direction, Δt_1 is the total time of the avoidance and r_h is the radius of human active area. Finally, the radius of the human zone r_0 should satisfy the constraint

$$r_0 \geq \bar{\delta}_{hx} + \bar{\delta}_{mrx1} \qquad (5)$$

where $\bar{\delta}_{hx}$ and $\bar{\delta}_{mrx1}$ is the human and robot moving distance in X-direction, respectively. Based on the above model, calculation for the worst case, gives the radius of the human zone of 30cm.

IV. Velocity potential field method with the consideration of human presence

In this paper, the velocity potential field method [3], obtained from velocity potentials defined in hydrodynamics, is modified for planning a path which avoids collisions with the human and the obstacles, such that, during the avoidance, the robot will avoid the human before it starts to avoid the obstacles. The robot is guided by its velocity commands which are given by its navigation controller (Figure 3).

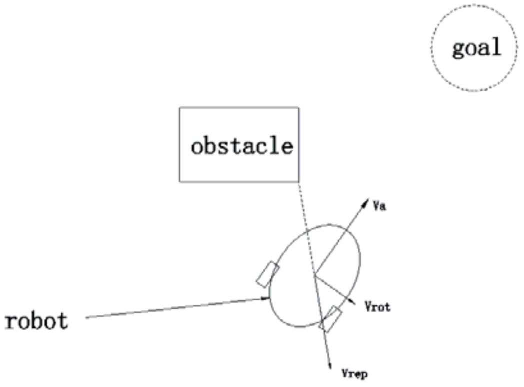


Figure 3. The Attractive, Rotation and Repulsive Velocity Commands

The distance between robot and goal d_{goal} is

$$d_{goal} = \sqrt{(yG - y)^2 + (xG - x)^2} \qquad (6)$$

where (xG, yG) refers to the goal and (x, y) to the center and robot.
The goal radius is

$$g_{rad} = 0.4m$$

VPF approach, presented in detail in [3], results in velocity commands for the robot controller.
The attractive velocity function is defined as [3]

$$V_a = 2 \times \bar{V}_{mr} \times (1 - \ell^{-\frac{d_{goal}}{g_{rad}}}) \qquad (7)$$

where $\bar{V}_{mr}$ is the maximum velocity of robot.
The repulsive velocity function is defined as [3]

$$V_{rep} = 0.5 \times \frac{1}{d_{obs}} \times \ell^{-\frac{d_{obs}}{obs_{rad}}} \qquad (8)$$

where d_{obs} is the distance between the obstacle and the robot which is obtained by the sensors.
Here, we also created a repulsive velocity function for the human, which has a larger gain than the repulsive velocity function for obstacle

$$V_{rephuman} = 0.7 \times \frac{1}{d_{obs}} \times \ell^{-\frac{d_{obs}}{obs_{rad}}} \qquad (9)$$

Both $\frac{1}{d_{obs}}$ and $\ell^{-\frac{d_{obs}}{obs_{rad}}}$ in the equation cause a sustained increase of the repulsive velocity if the robot keeps approaching the obstacles or human, which means that the closer the obstacle or human is, the larger is the repulsive velocity.
The rotation velocity function is defined as [4]

$$V_{rot} = 0.5 \times \frac{1}{d_{obs}} \times \ell^{-\frac{d_{obs}}{obs_{rad}}} \qquad (10)$$

where d_{obs} is the distance between obstacle and robot which is obtained by the sensors.

The resultant velocity command is given by

$$V_{sum} = \begin{cases} V_a, P_r \in A \\ V_a + V_{reph} + V_{roth} & P_r \in A, P_h \in A_0 \\ V_a + V_{reph} + V_{roth} & P_r \in A, P_h \in A_0, P_h \in A_1 \\ V_a + V_{rep} + V_{rot} & P_r \in A, P_{obs} \in A_1 \end{cases} \qquad (11)$$

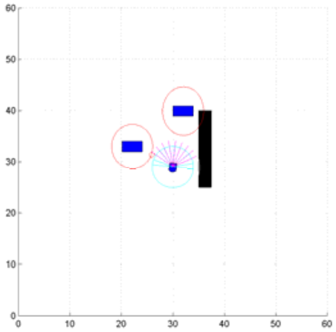
where $V_{a,}$, $V_{rep,}$ and $V_{rot,}$ are the attractive, repulsive and rotation velocity commands, respectively. The $V_{reph,}$ and V_{roth} are velocity commands regarding a human. A is the whole map area while A_0 and A_1 represent the human zone and obstacle zone.

V. Simulation results for human friendly robot

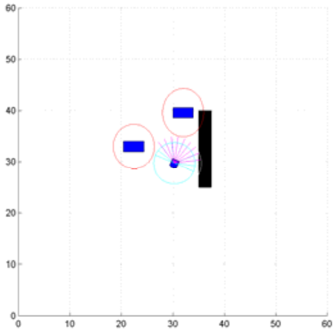
In order to test if the robot is able to avoid collisions as expected, simulations were carried out using MATLAB^{TM}. Figure 4 presents the results of the simulation with the following symbols:

The blue rectangle with the red circle: the human
The black rectangle: the obstacle
The blue polygon with light blue circle: the robot

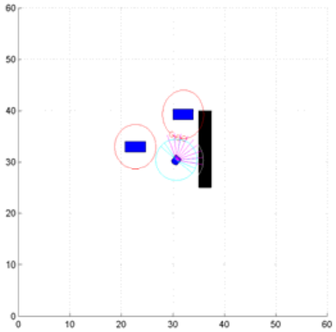
In these simulations, the robot did not know it will face a dangerous situation. The aim of creating such an extreme situation is to test if the robot controller has the ability to avoid the human and the obstacle while it still has time and space.



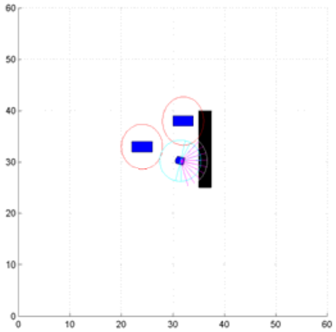
(a)



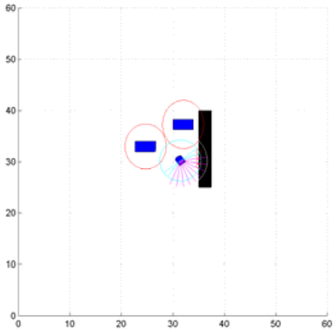
(b)



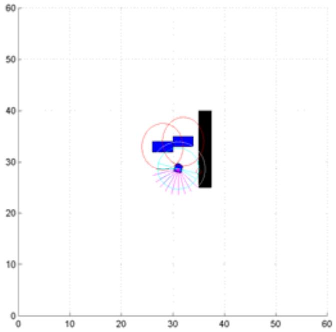
(c)



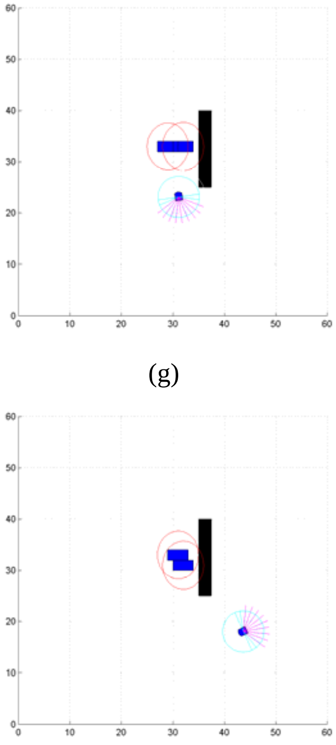
(d)



(e)



(f)



(g)

(h)

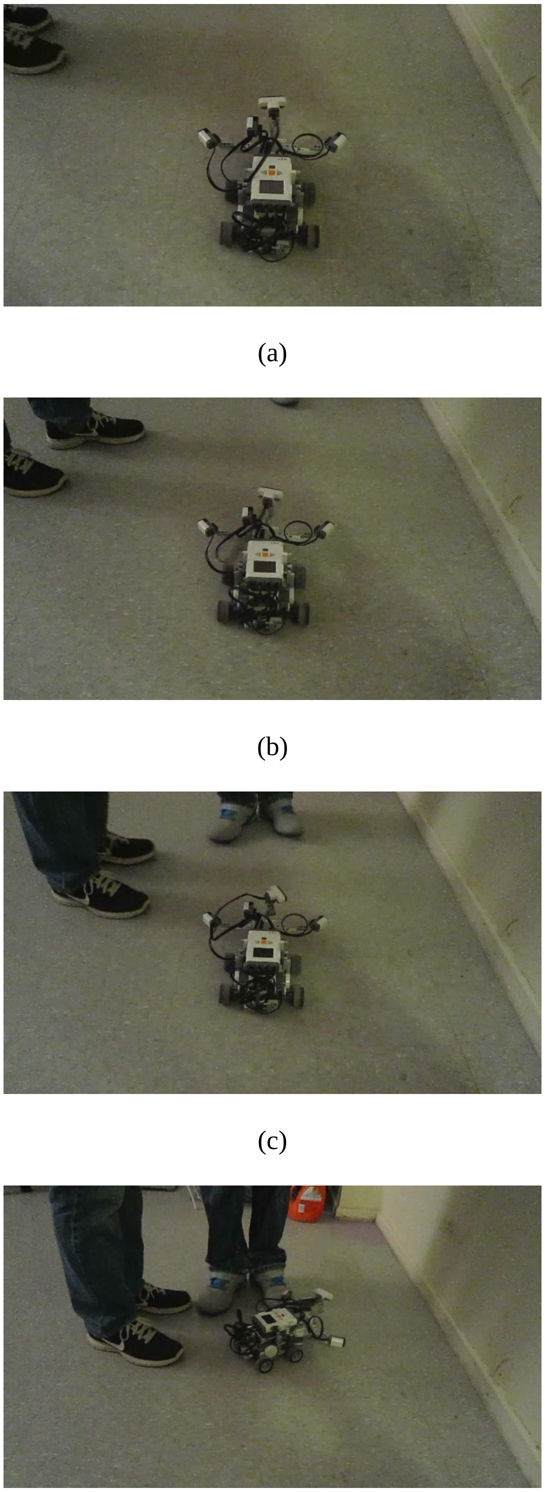
Figure 4. Higher Human Priority (Two Humans and One Obstacle)

As shown in Figure 4, the robot turned around and ran away from the two human and, afterwards avoided the obstacle moving in a direction where there is no danger. This illustrated that the proposed VPF based controller permits to avoid fixed obstacles and avoid humans and arrive to the designated goal.

VI. Experimental Results

Experiments were carried out for the same scenario as in simulations, in which the robot suddenly faces the moving human and obstacles at the same time and, during the avoidance, the robot should detour human and obstacles with different priorities, such that the human always has a higher priority in avoiding collision than obstacles.

For this scenario, two humans kept approaching the robot located in front and left side and the wall was in its right side. The purpose to design this situation is to test if the robot is able to avoid humans first (even has a risk to collide with an obstacle) and only afterwards to start avoiding the obstacles. As we can see from the Figure 5, the robot avoided the two humans first even while moving in the direction of the wall. Then, the robot avoided the wall and ran away from the danger of collision. It can be noted that the simulation results from Figure 4 (a)-(f) correspond to the snapshots of experimental results from Figure 5 (a)-(f). Experimental results confirmed that the proposed combination of Dempster-Shafer sensor fusion with velocity potential field based controller successfully avoided fixed obstacles and humans in moving towards the designated goal.

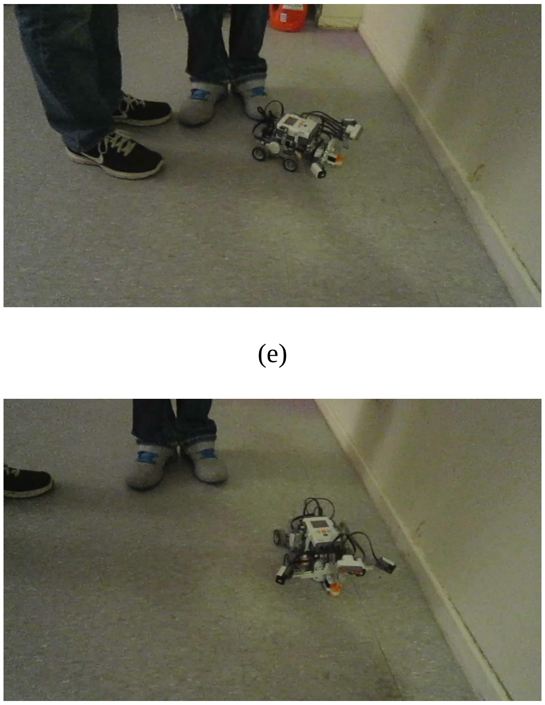


(a)

(b)

(c)

(d)



(e)

(f)

Figure 5 Two Human and One Obstacle Avoidance

VII. CONCLUSIONS

The paper presents a novel approach to robot-human collision avoidance using Dempster-Shafer method based on evidence theory. This methodology allows to integrate signals from multiple sensors. This supplies more reliable distance estimation to a velocity potential field controller.

The results presented in this paper show that the new human friendly mobile robot navigation controller based on Dampter-Shafer sensor fusion is able to lead the robot, avoid collisions with the obstacles and humans while always maintaining human safety as its first priority.

In the extreme case, when the robot does not have enough room to avoid the collision with the human and an obstacle, it chooses to protect the human even if in the process it might collide with an obstacle.

In the work presented in this paper, the robot has only one infrared sensor to sense the human. Although the sonar sensor helps the infrared sensor it cannot distinguish human on its own. This method is good enough for human detection, but it cannot ensure complete security for the human. In the future, another sensor based on infrared camera will have to be added to the robot so that human movements can be further recorded by the robot through processing pictures captured by the infrared camera and thus the human safety can be further improved.

REFERENCES

[1] G. Benet, F. Blanes, J. E. Simó, and P. Pérez, “Using infrared sensors for distance measurement in mobile robots,” Rob. Auton. Syst., vol. 40, no. 4, Sep. 2002, pp. 255–266.

[2] G. Tuna, V. C. Gungor, and K. Gulez, “An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters,” Ad Hoc Networks, vol. 13, Feb. 2014, pp. 54–68.

[3] E. Pruner, “Control of Self-Organizing and Geometric Formations”, A thesis presented for the degree of Master of Applied Science in engineering, University of Ottawa, 2013, pp. 66-73.

[4] L. A. Klein, “Sensor and data fusion: a tool for information assessment and decision making”, Spie.org, 2004, pp. 149-180.

[5] R. R. Murphy, “Dempster–Shafer Theory for Sensor Fusion in Autonomous Mobile Robots”, IEEE Transactions on Robotics and Automation, vol.14, No.2, Apr 1998, pp. 197-206.

[6] L. Zeng, “Design and control of human friendly robots”, A thesis presented for the degree of Ph.D of Applied Science in engineering, McMaster University, 2010, pp. 79-92.

[7] H. M. Choset, “Principles of robot motion: theory, algorithms, and implementations.” MIT press, 2005, pp. 301-322.

[8] G. M. Bone and L. Zeng,“Collision avoidance for nonholonomic mobile robots among unpredictable dynamic obstacles including humans,” 2010 IEEE Int. Conf. Autom. Sci. Eng, Aug. 2010, pp. 940–947.

[9] Y. Lu, L. Zeng and G. M. Bone, “Multisensor System for Safer Human-Robot Interaction,” 2005 IEEE International Conference, April, 2005, pp. 1767–1772.

[10] G. Feng, X. Guo and G. Wang, “Infrared motion sensing system for human-following robots,” Sensors Actuators A Phys., vol. 185, Oct. 2012, pp. 1–7.

[11] M. Kam, X. Zhu and P. Kalata, “Sensor Fusion for Mobile Robot Navigation,” Proceedings of the IEEE 1997, vol. 85, no. 1, 1997, pp108-119.

[12] T. Ali and P. Dutta, “Methods to Obtain Basic Probability Assignment in Evidence Theory”, Int. J. of Computer Applications, Jan. 2012, pp. 46-51.

[13] Y. Guo, A. Song, J. Bao, H. Tang and J. Cui, “A Combination of Terrain Prediction and Correction for Search and Rescue Navigation”, Int. J.of Advanced Robotic Systems, No. 3, 2009, pp. 207-214.

[14] Y. Hu, “Human Friendly Robot”, MASc Thesis, University of Ottawa, 2014, pp. 2-14.

Enhancing Data Security in Cloud Computing Using a Lightweight Cryptographic Algorithm

Sana Belguith
Laboratory of Electronic Systems and Communication Network, Tunisia Polytechnic School
Telnet Innovation Labs, Telnet Holding
E-mail: sana.belguith@telnet-consulting.com

Abderrazak Jemai
Laboratoire LIP2, Faculté des Sciences de Tunis, Tunisie
E-mail: Abderrazak.Jemai@insat.rnu.tn

Rabah Attia
Laboratory of Electronic Systems and Communication Network, Tunisia Polytechnic School
E-mail: Rabah.attia@enit.rnu.tn

Abstract— **Cloud computing is a new architecture that has released users from hardware requirements and complexity. The rapid transition toward clouds has advanced many concerns related to security issues which can hold back its widespread adoption. In fact, cloud computing's special architecture has introduced many challenges especially in maintaining the security of outsourced data. Thus, to address this issue, we propose in this article, a new lightweight encryption algorithm which consists of combining symmetric algorithm to encrypt data and asymmetric one to distribute keys. This combination helps to benefit from the efficient security of asymmetric encryption and the rapid performance of symmetric encryption while conserving the rights of users to access data by a secured and authorized way. Evaluation results prove that the processing time of our lightweight algorithm is faster than state-of-the-art cryptographic algorithms.**

Keywords-Cloud computing; Security; Privacy; Data security; Cryptography.

I. INTRODUCTION

In recent years, there has been a huge proliferation of the distributed computing systems use and advancement. This increase has produced a large amount of network distributed paradigms, infrastructures and architectures such as Grid, Pervasive, Autonomic, Cloud, etc.

Cloud computing refers to a network of computers, usually connected through internet, sharing an amount of resources scalable to reach the user's needs and offered by a service provider [1].

The US National Institute of Standards and Technology (NIST) [2] defines cloud computing as follows: '*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*'.

Cloud computing allows users to access software applications and computing capabilities, while using different service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [3]. These three service models are described below:

- Infrastructure as a Service (IaaS) enables the consumer to provide fundamental computing resources (such as processing, storage, networks, etc.). The consumer can deploy and run different kinds of software including operating systems.
- Platform as a Service (PaaS): This model enables the consumer to deploy onto the cloud infrastructure applications created or acquired by the consumer.
- Software as a Service (SaaS): In this model, the user can benefit of the capability of using applications already deployed on the cloud environment by a provider.

Four deployment models are used to deploy cloud computing solutions. The first model is the private cloud which is a cloud infrastructure available only for an exclusive use by a single organization. It is managed by the organization itself or by a third party. The second model is the public cloud providing the cloud infrastructure to the use of the general public. It is owned and managed by a third party who is the cloud provider. The third cloud deployment model is the community cloud which consists of several organizations, having the same interests such as security requirements, share the cloud infrastructure. The last model is the hybrid cloud composed of two or more cloud infrastructures (private, community, or public), which are independent but associated, by standardized or proprietary technology, in order to reach information portability.

On the other hand, there are currently several challenges facing cloud computing mainly related to scalability, interoperability and multi-tenancy. But, the most important issues are related to the security since cloud computing as a system using internet network (such as grid computing, embedded systems, etc.) is exposed to a number of attacks [4]. The cloud computing security issues can hold back its widespread adoption. In fact, sharing resources in cloud computing causes the problem of maintaining these resources secured and protected from malicious access or use. This problem is particularly faced on the data outsourced to the cloud.

In order to provide a novel mechanism that enhances data security in the cloud computing environment, this paper introduces a new lightweight cryptographic algorithm useful to encrypt data outsourced to cloud storage. This proposed solution is based on the combination of symmetric and asymmetric cryptography to encrypt data. This combination helps to benefit from the efficient security of asymmetric encryption and the rapid performance of symmetric encryption while conserving the rights of users to access data by a secured and authorized way. The paper introduces a comparison of the two categories of encryption algorithms (Asymmetric and symmetric) using various input files. Then, it offers an evaluation of the proposed algorithm.

The rest of the paper deals with the following points. First, Section II surveys the security problems addressed by cloud computing. After a literature review and a description of existent solutions in Section III, we provide a logical description of the proposed algorithm in Section IV. Then, in Section V, we present a comparative study of a variety of cryptographic algorithm and we implement the new algorithm. Finally, Section VI proposes some final remarks and future works.

II. Cloud computing security

A. Trust

Trust is defined as "*the act of having confidence and reliance on someone or something to behave as promised*" [5]. In computer science, trust goes through many areas, such as security and access control in computer networks, reliability in distributed systems, etc. [6].

The fact of outsourcing data and applications, in a cloud environment, delegates their control out of the owner's strict control to the cloud provider. As a consequence, Trust depends on the deployment model and on the cloud provider.

B. Cloud security issues

Due to the novel architecture of cloud computing, many traditional security issues are countered effectively. Although, its infrastructure's singular characteristics have introduced a number of distinctive security challenges.

Security in general is related to the AIC triad, namely, Availability, Integrity and Confidentiality. These three properties have become the key aspects used in designing secure systems, especially, in the case of cloud computing architecture.

1) Confidentiality: it refers only to authorized parties or systems having the ability to access protected data [6]. Outsourcing data, delegating its control to a cloud provider and making it accessible to different parties increase the risk of data breach. A number of concerns emerge regarding the issues of multi-tenancy, data remanence, application security and privacy [7]. Multi-tenancy refers to the cloud characteristic of resource sharing [6]. The cloud computing architecture consists of sharing different kinds of resources to enable multiple clients to use the same resource at the same time which presents a number of privacy and confidentiality threats.

2) Integrity: It means that only authorized parties can modify assets in authorized ways and it refers to data, software and hardware. Data Integrity refers to protecting data from unauthorized deletion, modification or fabrication [6]. Authorization is the mechanism used by the system to determine what level of access a particular authenticated user should have to secure resources [6]. Due to the rise of the number of parties involved in a cloud environment, authorization is important to ensure data integrity.

3) Availability: It refers to the property of a system being accessible and usable upon demand by an authorized entity. System availability includes a system's ability to carry on operations even when some authorities misbehave [6]. To ensure availability, the system has to be able to operate even if there is a security threat. The user of a cloud environment, who is discharged of hardware infrastructure requirements, relies on the availability of the ubiquitous network.

III. Literature review

Information security is defined by IEEE as "*the degree to which a collection of data is protected from exposure to accidental or malicious alteration or destruction*" [8].

The International Standard Organization in 2005 defines information security by the ISO/I EC 27002 standard. It is specified by the standard as "*the preservation of confidentiality (ensuring that information is accessible only to those authorized to have access), integrity (safe guarding the accuracy and completeness of information and processing methods) and availability (ensuring that authorized users have access to information and associated assets when required)*" [9].

To achieve data security, many techniques are used. These techniques can be categorized into four categories: masking, erasure, backup, and encryption. In an unreliable environment, like cloud computing, where there is not a physical control over our data, cryptography seems to be the best way to secure outsourced data. In fact, multi-tenancy characteristics and easy provider access to data force us to ensure confidentiality through innovative techniques based essentially on encryption techniques and access control.

Data encryption refers to the technique that uses cryptography theory to encrypt data on storage devices. Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [10]. Cryptography systems are classified into two basic types: Symmetric techniques (also known as conventional or secret key) and asymmetric techniques (public key).

Symmetric encryption consists of using the same key by the sender and the receiver parties. This key is used once to encrypt the data by the sender, and then, time it is used by the recipient to decrypt the data. The asymmetric encryption is based on the use of a pair of keys: a Public key and a private key. The data are encrypted by the public key which can be published. Then, the data are decrypted by the private key which must be kept secret.

In 2012, Dimitrios et al. [6] studied in depth cloud security while identifying security requirements. The authors proposed the use of a trusted third party as a security solution. The solution consists of using cryptography, specifically public key infrastructure with Single-Sign-On (SSO) mechanisms [11] and the lightweight directory access protocol (LDAP) [11], to ensure access control, data integrity and confidentiality and secured communications. This proposed solution consists of using cryptography to ensure confidentiality and integrity of involved data. However, it does not identify encryption algorithms to be used.

In 2009, Cunsolo et al. [12] proposed a mechanism to protect data in distributed systems (grid, cloud, autonomic, etc.). This technique consists of the use of a combination of symmetric and asymmetric cryptographic algorithms. Although in this scheme, only the data owner can access the data which contradicts the concept of sharing resources in the cloud environment.

Hashizume et al. [13] presented a classification of security issues in different service models (SaaS, PaaS and IaaS). This paper offers an identification of the main vulnerabilities in cloud computing while presenting the common threats and its relations to cloud layers. In spite of proposing some available countermeasures, this study does not provide technical implementation of these solutions.

In 2013, Rahmani et al. [14] proposed Encryption as a Service (EaaS) as a solution for cryptography in cloud computing based on XaaS concept. This solution presents a response to prevent the security risks of cloud provider’s encryption and the inefficiency of client-side encryption. However, there is not a comparative study of cryptographic algorithms which can be integrated in this solution.

Bugiel et al. [15] proposed a secured cloud architecture which consists of using two clouds (twins) to perform computations. This model differentiates between computations based on their security and performance aspects; trusted cloud performs security-critical operations and the commodity cloud performs performance-critical operations.

In [16], Mohammad et al. evaluated the performances of cryptographic algorithms (symmetric and asymmetric algorithms) in a cloud platform. Based on key size, the performance and the size of the output file, the paper offers a study of different encryption techniques in a cloud environment. Finally, it proposes to use AES algorithm to encrypt data but it does not propose a secure way to distribute encryption keys.

Dimitrios et al. was the first proposing the use of cryptography to secure cloud architecture [6]. Ever since, many authors proposed to use cryptographic algorithms in the cloud storage [14][16]. But, these solutions remain incomplete because they do not specify which algorithm is recommended to encrypt data and how to distribute cryptographic keys while maintaining adequacy with cloud characteristics.

IV. Methodology

In this section, we provide a logical description of the approach proposed to countermeasure data security issues in cloud computing.

A. *Proposed Solution for enhancing data security in the cloud*

While using cloud storage, sharing resources, especially sharing data between data owner and authorized clients, can pose the risk of data breach or leakage. In fact, securing data in the cloud is difficult to fulfill if the client does not trust the service provider. The client is obliged to blindly trust the provider’s mechanisms but this can be hold back by the threats of malicious insiders among them cloud administrators who can access data simply.

To ensure data security, many techniques and technologies were proposed among them the most efficient one is cryptography. The most successful approach adopted to secure data is symmetric cryptographic techniques. But, this technique alone is not efficient in a multi-tenant scenario; many authorized clients have the right to access data so the key has to be distributed to each client. The key management is too difficult to perform since there are risks in sending keys to different clients at the same time.

The asymmetric cryptographic techniques could be a suitable way to ensure data security. Although, this solution restricts data access to the data owner which contradicts the multi-users aspect of cloud computing. Besides, the asymmetric cryptography algorithm presents a heavy impact on data access. This usually does not allow the encryption of large amounts of data in acceptable time for users. Thus, it is necessary to propose a new solution that can offer a trade-off between security requirements and system performance and which preserves the multi-tenancy aspect of cloud computing.

The solution we propose combines the efficient security of asymmetric encryption and the rapid performance of symmetric encryption while conserving the rights of users to access data by a secured and authorized way. In our model, the data will be encrypted by a symmetric algorithm. Then, the symmetric key distribution between cloud provider and authorized users will be performed using asymmetric algorithm. Our new solution is basically proposed to protect data confidentiality and integrity using encryption. Moreover, the cryptographic keys are stored at cloud provider side, to decrypt/encrypt data and share it with other clients. The data stored in the cloud remain available on demand by authorized clients and can be accessed rapidly.

B. *Enhancing data security's algorithm*

The data in the cloud storage are encrypted by a symmetric key cryptographic algorithm.

1) Key exchange

The first step of the algorithm is the key exchange. This step is composed of two phases: key generation and key exchange.

The data owner generates the symmetric key used to encrypt data, and then, asymmetric two keys used to encrypt the symmetric key. He sends his public key to the cloud storage, which, in turn, generates his two asymmetric keys. Then, the cloud storage saves the cloud public key *Cpub* and the owner's public key *Upub*. In the second phase, the data owner requests the public key of the cloud provider and he encrypts the symmetric key *ksym* with the public key of the cloud provider. Finally, he sends K_E to the cloud storage to be stored.

The step by step algorithm describing the key exchange phase is reported by the diagram of Figure 1.

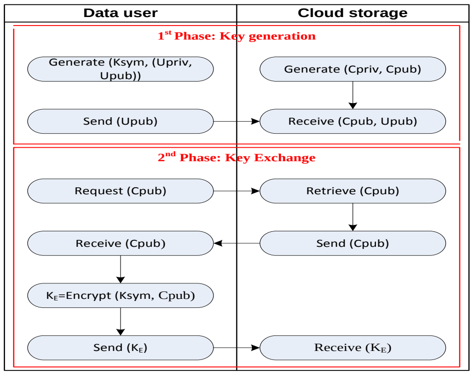


Figure 1. Key generation phase.

2) Data storage

The second step of the algorithm is the storage of the encrypted data in the cloud storage. In this step, the data owner encrypts his data by the symmetric key *ksym* and he sends it to the cloud storage in encrypted form. This phase is described in Figure 2.

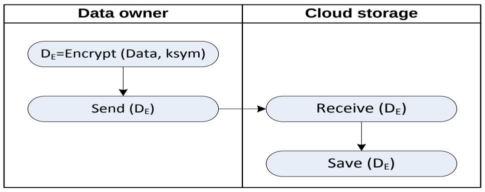
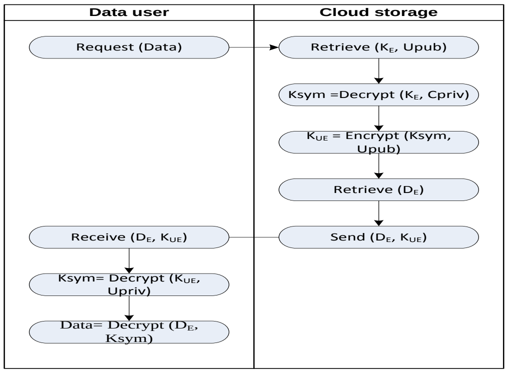


Figure 2. Data storage phase.

3) Data access

To access the data stored in the cloud storage, the user requests the data from the cloud storage as shown in Figure 3.



Data user
Cloud storage
Request (Data)
Retrieve (K_E, Upub)
Ksym =Decrypt (K_E, Cpriv)
K_{UE} = Encrypt (Ksym, Upub)
Retrieve (D_E)
Receive (D_E, K_{UE})
Send (D_E, K_{UE})
Ksym= Decrypt (K_{UE}, Upriv)
Data= Decrypt (D_E, Ksym)

Figure 3. Data access phase.

The cloud storage finds the encrypted symmetric key K_E and the user public key. Then, the cloud storage decrypts the symmetric key K_E with his own private key *Cpriv* and re-encrypt it with the user public key *Upub*. Next, he finds the encrypted data and he sends it with the encrypted key to the user. The user, in turn, decrypts the symmetric key with his private key *Upriv* and he decrypts the data with the symmetric key.

V. IMPLEMENTATION OF THE SOLUTION IN THE CLOUD ENVIRONMENT: RESULTS AND DISCUSSION

A. *Experimental environment*

In order to study the performance of the solution theoretically proposed in the previous section, we have implemented different kinds of symmetric and asymmetric cryptographic algorithms in the cloud environment. Then, we have implemented the hybrid algorithm proposed.

The experimental environment consists of the cloud network composed by the hypervisor Xen Server (6.1), the middleware Openstack and the client that uses Citrix Desktop [17] to access to the virtual machine hosted by XenServer. The cloud server uses the Core I5 (4.8 GHz) with 4 GB of RAM and the client machine utilizes the Core I3 with 4 GB of RAM.

B. *Experimental results and discussion*

In this section, the study distinguishes between symmetric and asymmetric algorithms by implementing them in the same cloud environment [18].

In Table I, all the symmetric algorithms such as AES [10][19], DES [10], 3DES [10] and Blowfish [10], and asymmetric techniques like RSA [10] and ElGamal [10], have been implemented and tested using different input text file sizes: 1 MB, 10 MB, 50 MB and 100MB. The results mentioned in the table I, are the average of the encryption time calculated after doing the experiments three times.

TABLE I. PROCESSING TIME AND KEY SIZE

		AES	**DES**	**3 DES**	**Blowfish**	**RSA**	**ElGamal**
Key size		256	64	192	256	2048	1024
File size (MB)	1	0,03	0,03	0,09	0,03	332.29	2935.90
	10	0,32	0,32	0,77	0,24	-	-
	50	1,61	1,89	4,49	2,13	-	-
	100	4,27	5,50	7,96	5,64	-	-

From this analysis, we can conclude that the symmetric algorithms are faster than the asymmetric algorithms. However, the asymmetric algorithms are the most robust against the code breaking thanks to the lengthy keys used. Among the symmetric algorithms, AES has the lowest data processing time. Moreover, RSA is the asymmetric algorithm having the longest key size as shown in Figure 4.

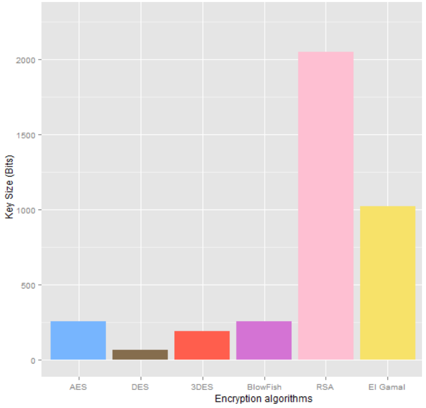


Figure 4. Key size of different cryptographic algorithms.

Obviously, Figure 5 represents the processing time of the implemented symmetric and asymmetric algorithms in the cloud environment.

Finally, using the analysis above, we choose to evaluate the new algorithm proposed by combining the AES algorithm as the symmetric technique and the RSA algorithm as the asymmetric technique.

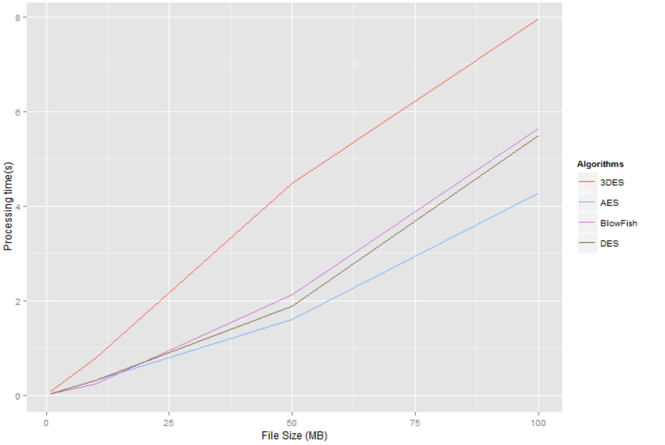


Figure 5. Processing time of different cryptographic algorithms.

TABLE II. PROCESSING TIME OF THE NEW HYBRID TECHNIQUE

File size (MB)	**1**	**10**	**50**	**100**
Time (s)	0,06	0,30	1,93	4,02
	0,02	0,22	1,54	2,89
	0,03	0,19	1,08	2,83
Average time (s)	0.04	0.24	1.52	3.25

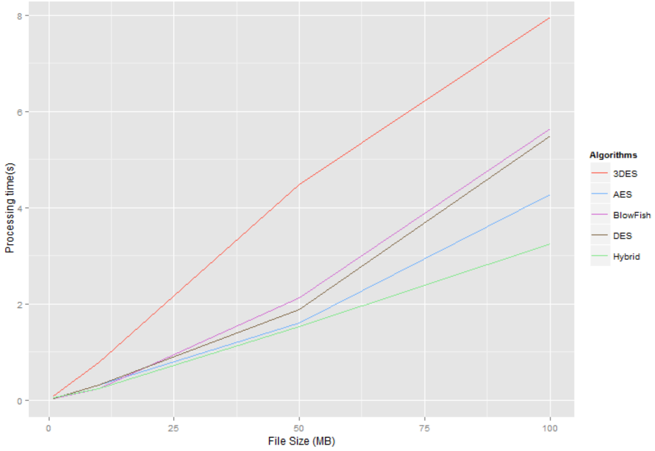


Figure 6. Processing time of the new hybrid technique.

The data processing time composed by the encryption and key distribution time of the new algorithm is faster than the other algorithms while having the secure key distribution as shown in Table II and Figure 6. By this hybrid technique, we can achieve a rapid performance of data processing and an efficient security level using the key distribution mechanism. This new technique enhances confidentiality and integrity of data stored in the cloud while maintaining it available on-demand.

VI. CONCLUSION AND FUTURE WORK

Data security is an open problem in cloud computing environment. To ensure data security, many techniques and technologies were proposed among them the most efficient one is cryptography.

In this work, we propose a hybrid encryption technique consisting of using asymmetric and symmetric cryptographic algorithms. In our model, the data is encrypted by a symmetric algorithm. Then, the symmetric key distribution between cloud provider and authorized users is performed using an asymmetric algorithm.

This paper offers a comparison of the two categories of encryption algorithms (Asymmetric and symmetric) using various input files. Based on the output analysis, we can conclude that the symmetric algorithms are the more efficient in cloud environment thanks to the rapidity of processing data, among them the AES is the faster one. Moreover, the analysis proves that the asymmetric algorithms are more robust thanks to the key length used especially the RSA algorithm. Finally, we evaluated the hybrid technique using the combination of the AES algorithm and the RSA algorithm. This analysis proves that the novel technique enjoys the advantages of symmetric algorithm in the processing time and the robustness of asymmetric algorithm in key length. In fact, this new lightweight algorithm is faster than other cryptographic techniques in processing data. Moreover, it is robust and secured thanks to its key distribution mechanism.

This work can be enhanced by proposing a new key distribution scheme which aims to give every authorized user the encryption key without the cloud provider interaction.

ACKNOWLEDGMENT

This work is a part of the MOBIDOC project achieved under the PASRI program, funded by the European Union and administered by the ANPR [20].

REFERENCES

[1] M. Koehler and S. Benkner, "VCE-A Versatile Cloud Environment for Scientific Applications." The Seventh International Conference on Autonomic and Autonomous Systems (ICAS'11) IARIA, 2011, pp. 81-87

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, Information Technology Laboratory, 2011.

[3] A. Tchana, L. Broto, and D. Hagimont, "Fault Tolerant Approaches in Cloud Computing Infrastructures", The Eighth International Conference on Autonomic and Autonomous Systems ICAS'12), 2012, pp. 42-48.

[4] A. Jemai, A. Mastouri, and H. Elleuch, "Study of key pre-distribution schemes in wireless sensor networks: case of BROSK (use of WSNet)", International Journal of Applied Mathematics & Information Sciences (AMIS'12). 2012, pp. 655-667.

[5] P. I. Bhosle and S. A. Kasurkar "Trust in Cloud Computing". International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 2, Issue 4, 2013, pp. 1541-1548.

[6] D. Zissis and D. Lekkas. "Addressing cloud computing security issues". Future Generation Computer Systems, 28(3), 2012, pp. 583-592

[7] Cloud Security Alliance. Top threats to cloud computing, Cloud Security Alliance, 2010.

[8] The Authoritative Dictionary of IEEE Standards Terms, 7th ed., Institute of Electrical and Electronics Engineers, Los Alamitos, CA, USA, 2000.

[9] ISO/IEC 27002:2005 Standard: Information technology – Security techniques – Code of practice for information security management, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), Geneve, Switzerland, 2005.

[10] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, "Handbook of Applied Cryptography", Boca Raton, FL, USA:CRC Press, Inc., 1996.

[11] "Directories and Public –Key Infrastructure (PKI)", VeriSign, 2004.

https://www.verisign.com.br/static/Directories_PKI.pdf

[12] V. D. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa, "Achieving information security in network computing systems", Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'09.), 2009, pp. 71-77.

[13] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing". Journal of Internet Services and Applications, vol. 4, 2013, pp. 1-13.

[14] H. Rahmani, E. Sundararajan, Z. M. Ali, and A. M. Zin, "Encryption as a Service (EaaS) as a Solution for Cryptography in Cloud". Procedia Technology, vol. 11, 2013, pp. 1202-1210.

[15] S. Bugiel, S. Nürnberger, A. R. Sadeghi, and T. Schneider, "Twin clouds: Secure cloud computing with low latency". Communications and Multimedia Security. Springer Berlin Heidelberg, 2011, pp. 32-44.

[16] J. Mohammad, K. Omer, S. Abbas, E. S. M. El-Horbaty, and A. B. M Salem, "A comparative study between modern encryption algorithms based on cloud computing environment". 8th International Conference for Internet Technology and Secured Transactions (ICITST'13), IEEE, 2013, pp. 531-535.

[17] mc software company, "Cloud Lifecycle Management with CitrixXenServer Virtualization", manual guide, network partner, 2013.

[18] «PyCrypto - The Python Cryptography Toolkit». Available from: https://www.dlitz.net/software/pycrypto/

retrieved: 2015.03.10.

[19] A. Sachdev and M. Bhansali, "Enhancing Cloud Computing Security using AES Algorithm", International Journal of Computer Applications, vol. 67, No. 9, 2013, pp. 19-23.

[20] http://www.pasri.tn/pr%C3%A9sentation

retrieved:2015.04.06.

Supporting the Neon and VFP Instruction Sets in an LLVM-based Binary Translator

Yu-Chuan Guo, Wuu Yang, Jiunn-Yeu Chen
Computer Science Department
National Chiao-Tung University
Hsinchu, Taiwan, R.O.C.
Emails: qoo12345654321@gmail.com, {wuuyang, jiunnyeu}@cs.nctu.edu.tw

Jenq-Kuen Lee
Computer Science Department
National Tsing-Hua University
Hsinchu, Taiwan, R.O.C.
Email: jklee@cs.nthu.edu.tw

Abstract—**Binary translation attempts to emulate one instruction set with another on the same or different platforms. This important technique is widely used in instruction-set-architecture migration, binary instrucmentation, dynamic optimizations, software security, and fast arhitecture simulation. Vector and floating-point instructions are widely used in many applications, including multimedia, graphics, and gaming. Though these instructions are usually simulated with software in a binary translator, it is important to support them in such a way that the host SIMD (single instruction multiple data) and floating-point hardware is efficiently used in the translation process. We report our design and implementation of the emulation of ARM Neon and VFP (vector floating point emulation) instructions in the MC2LLVM (machine-to-low-level-virtual-machine) binary translator. Our approach can take full advantage of the vector and floating-point functional units, if present, of the host machine. The experimental results show that code generated by MC2LLVM with the Neon and VFP extensions achieves an average speedup of 1.174x in SPEC 2006 benchmark suites compared to code generated by MC2LLVM without the Neon and VFP extensions.**

***Keywords*–binary translation; cloud computing; LLVM; floating-point instruction; Neon, VFP; vector instruction; virtualization.**

I. Introduction

Binary translation [14] attempts to emulate one instruction set with another on the same or different platforms. The important technique is widely used in instruction-set-architecture migration [4][5][13][18], binary instrucmentation [6], dynamic optimizations [2][11], software security, and fast arhitecture simulation [15], [17]. The Neon and VFP coprocessors [1] are extensions to the ARM architecture. They are designed for applications with SIMD and floating-point instructions to meet the growing demand of computing power in embedded systems, such as multimedia, 2D/3D graphics, and gaming.

Existing binary translators, such as QEMU (quick emulator), support vector and floating-point instructions with naive software simulation (using scalar, integer, and shift operations) [3]. The result is poor performance. In this paper, we attempt to make use of the vector and floating-point hardware on the host platform, if present, to execute the Neon and VFP instructions. Furthermore, all the related exceptions and flags in the floating-point environment are taken care of properly. Our system is built on top of MC2LLVM, a retargetable static/dynamic/hybrid binary translator developed in our lab in the past few years. The resulting performance is the key issue in our design.

We leverage the existing LLVM backend (low-level virtual machine) to build a high-performance and retargetable binary translator. In this research, the emulation of Neon and VFP architectures is layered on the top of an LLVM backend.

To be fully compliant with Neon and VFP instruction set architectures, we need to know the details of the machine features [1] in the Neon and VFP architectures, including the flush-to-zero mode, the default NaN (Not a Number) mode, and the floating-point exceptions. These details can affect the behaviors of the Neon and VFP instructions. Our implementation must faithfully mimic these features. We also propose new methods to detect floating-point exceptions if the IR layer of a binary translator does not provide the relevant information.

To achieve a high degree of reliability, we also developed a verification framework for testing all the emulated instructions in Neon and VFP in order to gain confidence in the correctness of our approaches. Verification is performed automatically.

The remainder of this paper is organized as follows: Section 2 lists the related work and background knowledge, including Neon and VFP architectures, and describes the related terminology. In Section 3, we describe the implementation details. Section 4 illustrates the verification of the implementation. Section 5 discusses the experimental results. Section 6 concludes this paper.

II. Background and Related Works

In this research, we attempt to translate the Neon and VFP instructions into LLVM IR. In this chapter, we will introduce background knowledge and related works about the Neon and VFP architectures, including their machine features related to binary translation.

MC2LLVM (Machine Code To LLVM) is a process-level binary translation system based on LLVM developed in our lab in the past few years. It adopts the approach of the modern compiler techniques which separate the translation process into a frontend and a backend. The frontend translates the guest binary into LLVM IR and then uses the existing LLVM backend to generate the host binary from LLVM IR.

Figure 1 shows the flow of dynamic binary translation in MC2LLVM. The emulation manager maintains and manipulates the progress of the emulation, such as handling the control transfer between translated basic blocks in code cache and invoking the translator to translate a target basic block that has not yet been translated. In this research, we will focus on the translator module, which is also employed in the static and hybrid modes.

QEMU [3] is an open source binary translation system which supports full system emulation and, unlike MC2LLVM, always runs in the DBT mode. QEMU has its own IR, known as Tiny Code Generator (TCG) [3], to implement a two-stage

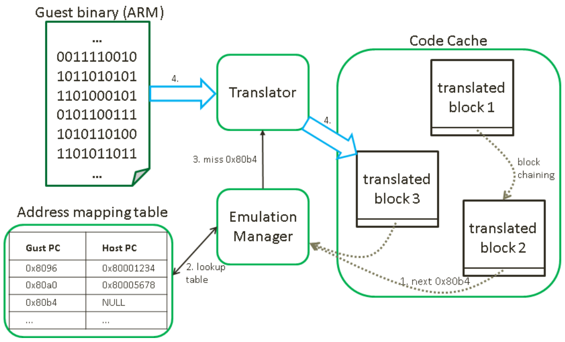


Figure 1. Structure of MC2LLVM.

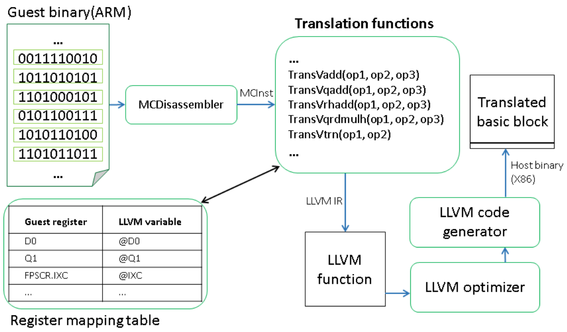


Figure 2. Translation from guest binary to host binary.

translation. It is able to emulate several ISAs, such as x86, PowerPC, ARM, and Sparc, etc.

Neon is a SIMD (Single Instruction Multiple Data) processor integrated into the ARM chip. Neon provides a 64/128-bit SIMD instruction set that provides acceleration for multimedia and signal processing applications, such as compressed video decoding, image processing, 2D/3D graphics, sound synthesis, etc.

VFP is a vector floating-point processor integrated as a part of the ARM chip. VFP provides a single-precision and double-precision floating-point instruction set that is fully compliant with IEEE 754 [9]. (Neon is not fully compliant with IEEE 754.)

Neon and VFP share the same extension register bank, which is distinct from the ARM core register bank.

The Neon and VFP extensions have a shared register space for system registers. Only the system register known as Floating-Point Status and Control Register (FPSCR) in this space is accessible in the application programs. FPSCR contains the arithmetic status flags as well as the bit fields for controlling the floating-point unit. Table I shows the bit fields of the FPSCR register.

The DN bit controls the default NaN (not-a-number) mode, which affects the behavior of the floating-point operations involving one or more NaNs. The NaN processing follows the IEEE 754 standard if the DN bit is 0. A floating-point operation involving one or more NaNs returns the default NaN if the DN bit is 1.

The FZ bit controls the flush-to-zero mode, which affects the behavior of the floating-point operations. If the FZ bit is 0, the behavior of a floating-point operation in VFP follows the IEEE 754 standard. If the FZ bit is 1, the flush-to-zero mode is enabled. In the flush-to-zero mode, denormalized numbers (in the IEEE 754 standard [9], denormalized or denormal numbers are very small numbers whose exponent fields are 0. They are used to fill the gap between zero and the minimum normalized number) will be flushed 0. The flush-to-zero mode also changes the criteria for the floating-point exceptions to occur (described in a later section). Neon always uses the flush-to-zero mode, regardless of the value of the FZ bit.

IDC, IXC, UFC, OFC, DZC, and IOC are input denormal, inexact, underflow, overflow, division by zero, and invalid operation cumulative exception flags, respectively. These flags show abnormalities during floating-point operations. A *cumulative* exception bit is set to 1 when the corresponding floating-point exception occurs. However, it is not reset to 0 when the corresponding exception does not occur automatically. These flags are usually used in applications with high safety requirements.

III. Design and Implementation

The implementation of the emulation of the Neon and VFP extensions, including the Neon and VFP registers, instructions, and machine features, is discussed in detail in this section.

Figure 2 shows the translation flow of MC2LLVM. The translator uses LLVM MCDisassembler [10] to disassemble binary instructions into MCInst (the IR of MCDisassembler). MCInst is translated by the functions we provided for each guest instruction into LLVM IR. The LLVM IR is organized as LLVM functions. The LLVM optimizer performs target-independent optimizations before code generation.

A. *Flush-to-Zero Mode Emulation*

The flush-to-zero mode [1] is a special processing mode that replaces denormalized operands, intermediate results, and final results with zero while reserving the sign bit. It is used to avoid handling denormalized numbers, thus saving execution time. Because supports for denormalized numbers increase the complexity in hardware design, floating-point units often save hardware cost by simply delegating supports for denormalized numbers to software.

The flush-to-zero mode is not compliant with IEEE 754. It also changes the behavior of a floating-point operation and the criteria for the floating-point exceptions in three ways: (1) In the flush-to-zero mode, a floating-point operation can cause the input denormal exception, which is not included in IEEE 754. (2) The inexact exception would not be raised when a result is rounded to zero or flushed to zero. (3) If the result of an operation is rounded to zero or flushed to zero, the underflow exception would be raised.

There are two operations in the flush-to-zero mode: flush-input-to-zero and flush-output-to-zero. Each floating-point operation must go through the preprocessing in LLVM IR in which denormalized numbers are detected and replaced with zero before a floating-point operation is performed. This is the flush-input-to-zero operation, shown in Figure 3.

According to the definition of the flush-to-zero mode, if the intermediate result of a floating-point operation that is

TABLE I. Floating-Point Status and Control Register format

bit(s)	31	30	29	28	27	26	25	24	
meaning	N	Z	C	V	QC	AHP	DN	FZ	
bit(s)	23:22	21:8	7	6:5	4	3	2	1	0
meaning	RMode	reserved	IDC	reserved	IXC	UFC	OFC	DZC	IOC

```
procedure FP32_FlushInputToZero (operand)
    if (IsDenormal(operand))
        SetIDC(); // input denormal exception
        flush operand;

function IsDenormal(op)
    if (abs(op) < 0x00800000 && abs(op) > 0)
        return TRUE;
    return FALSE;
```

Figure 3. Algorithm for flush-input-to-zero.

```
procedure FP32Mul_FlushOutputToZero(op1, op2, result)
    if (isDenormal(result))
        SetUFC(); // underflow exception
        flush result;
    else if (isZero(result) && !isZero(op1) && !isZero(op2))
        SetUFC(); // underflow exception
    else if (isMinNorm(result))
        multiply op1’s and op2s fractions with leading 1 to A;
        if( A has 23 consecutive 1s starting at the most significant 1)
            SetUFC();
            flush result;
```

Figure 4. Algorithm for flush-output-to-zero for single-precision floating-point multiplcation.

produced *before* rounding satisfies the following condition:

$$(1)\quad 0 < abs(result) < +MinNorm$$

where $MinNorm$ is the minimum normalized number of the destination precision, that intermediateg result is flushed to zero before the rest of the floating-point operation. However, an LLVM floating-point operation can only produce a result *after* rounding. The intermediate result produced after the floating-point operation but before rounding is invisible. Therefore, the flush-input-to-zero operation may produce an incorrect result if the intermediate result has already been rounded to $MinNorm$ or to zero. Hence the flush-output-to-zero operation, shown in Figure 4, is introduced to flush the intermediate and final results of a single-precision multiplication to zero.

If both operands of a floating-point multiplication are not zero but the result is zero, the result must have been rounded to zero. Obviously, the multiplication causes an underflow exception. Only when the intermediate result that is produced before rounding satisfies the following condition:

$$(2)\quad +MaxDenorm < abs(result) < +MinNorm$$

where $MaxDenorm$ is the maximum denormalized number (that is, the exponent field is all 0s and the fraction field is all 1s) of the destination precision, the result could be rounded to $MaxDenorm$ or $MinNorm$. In this case the after-rounding result may be a normalized number but it should be flushed to zero. We can reproduce the before-rounding intermediate result by (integer-)multiplying the two operands’ fractions. If the intermediate result has 23 (if single precision) consecutive 1s starting from the most significant 1, that is, condition (2) above, the result should be flushed to zero and an underflow exception should be raised.

Different floating-point instructions come with different emulations of the flush-output-to-zero operation. For example, for floating-point additions, it is not necessary to consider the two special cases (rounding to zero or to $MinNorm$) because a floating-point addition is always exact (i.e., without loss of precision) when the result is a denormalized number.

B. *Floating-Point Exception Emulation*

Floating-point exceptions are emulated with two methods. The first method is by checking the exception flags in the underlying hardware (based on C++11 Standard Library) and the second method is by employing additional test code. The second method is complete and more efficient than the first. We will explain both methods in this section.

Although LLVM supports many floating-point operations, it does not provide any information related to floating-point exceptions. The developers of LLVM may consider that accessing the floating-point environment is unlikely to happen and supporting them would diminish the performance because the implicit data dependencies that might occur in the floating-point environment. For example, setting floating-point exception flags or trapping the floating-point exceptions for further processing may unnecessarily restrict the LLVM optimizer and make the LLVM optimizer more complex. So we have to detect the floating-point exceptions with additional code.

The first method for detecting floating-point exceptions is to check the exception flags in the underlying hardware with the `fetestexcept` function in the C++11 Standard Library. Because the exception flags in the underlying hardware may be changed inadvertently by the emulation manager, translator, or other modules in MC2LLVM or LLVM during execution, it is necessary to clear the exception flags with the `feclearexcept` function before executing every floating-point operation in order to avoid these outside disturbances.

The first method comes with several drawbacks. First, MC2LLVM adopting this method becomes much slower than QEMU. We used a benchmark that repeats single-precision addition one billion times. The resulting execution time is shown in Figure 5. Furthermore, we wrote two new functions in x86

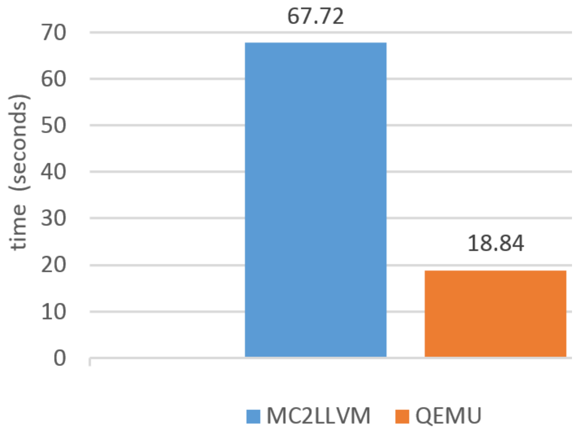


Figure 5. Execution time of single-precision floating-point addition. MC2LLVM adopts the first method for emulating floating-point exceptions.

assembly to replace `fetestexcept` and `feclearexcept` to make emulation of floating-point operation even faster; they are still a little (about 0.5 seconds) slower than QEMU evaluated by the benchmark. Second, the LLVM optimizer may reorder the instructions, which causes unexpected results.

The second method is to use additional code in LLVM IR to detect floating-point exceptions based on the operands and results of a floating-point operation. Normally, floating-point exceptions are detected at various instants during the execution of a floating-point operation. For example, the occurrence of the underflow and inexact exceptions is determined based on the before-rounding intermediate values of a floating-point operation. However, we cannot obtain any intermediate values from an LLVM floating-point operation. We could reproduce the intermediate values with software but this would incur high overhead. Alternatively, we use some tricks which we will discuss in details later for better efficiency, for example, employing other floating-point operations to test the occurrence of floating-point exceptions and bypassing useless detection of floating-point exceptions. We will use floating-point additions and multiplications to illustrate this approach. Other floating-point operations (`vcvt`, `vdiv`, `vcge`, `vcmpe`, etc.) are handled similarly. MC2LLVM adopting this second method is twice as fast as QEMU according to our benchmarks (for floating-point additions) mentioned earlier. Therefore, we decide to use this second method in our binary translator. In what follows we will discuss the emulation of each floating-point exceptions.

- Emulation of the Invalid Operation Exceptions
- Emulation of the Division by Zero Exceptions
- Emulation of the Overflow Exceptions
- Emulation of the Input Denormal Exceptions
- Emulation of the Inexact Exceptions
- Emulation of the Underflow Exceptions

Figure 6 summaries the emulation of a floating-point operation. Note that not all floating-point operations will go through all the steps shown in the figure. The emulation of floating-point operations involves many details and is error-prone for implementation. Therefore, a good verification technique is required for every implementation.

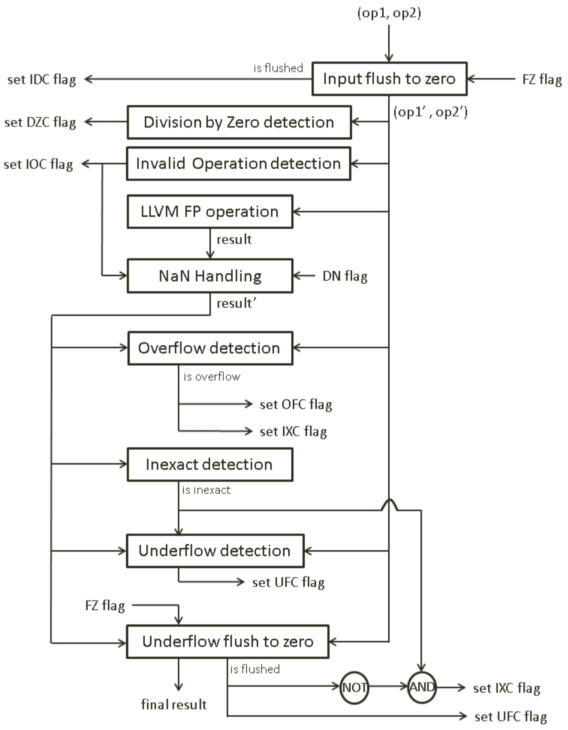


Figure 6. Emulation of a floating-point operation.

IV. Verfication of the Bianry Translator

Testing is a crucial building block in order to achieve a high degree of reliability. It is difficult to identify the mistranslated instructions generated by a binary translator because there are so many translation functions in a binary translator and, due to the complex interdependencies inherent in a floating-point operation, it is almost impossible to determine whether an instruction is functionally equivalent to its translated instructions directly.

Mistranslated instructions may produce wrong values, or fail to raise status flags such as the cumulative exception flags (`IXC`, `UFC`, etc.) and the cumulative saturation flag in FPSCR, in different combinations of modes, such as the flush-to-zero mode and the default NaN mode. Running a few benchmarks correctly is far from being correct for a binary translator because of low instruction coverage (an application usually makes use of less than 5% of the 1240 instructions in Neon and VFP), neglect of floating-point exceptions, and no mode switching in applications. Therefore, we developed a black-box testing framework for a binary translator, shown in Figure 7.

V. Experiments

In order to show the performance of our translation system, we compare the execution time of the guest binary with and without Neon and VFP instructions on MC2LLVM and QEMU, respectively. Since MC2LLVM can run in three different modes, we always use the pure dynamic translation mode in this experiments.

A. *Environment*

The experimental hardware is equipped with the Intel Core i7-4770 and 8GB memory, running 32-bit Ubuntu ver-

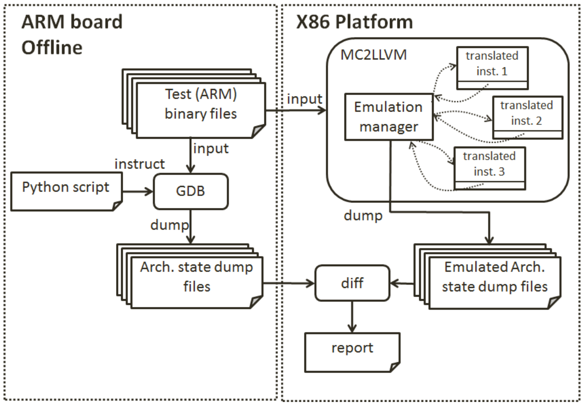
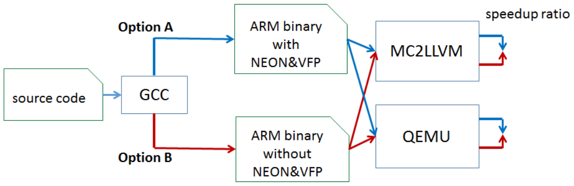
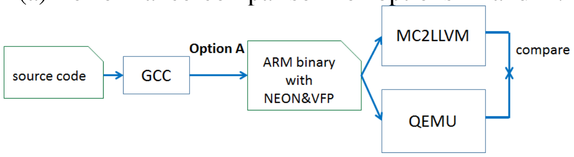


Figure 7. A verification framework.



(a) Performance comparison for options A and B.



(b) Performance comparison for MC2LLVM and QEMU.

Figure 8. Performance comparison.

sion 14.04. We use SPEC CPU2006 and LINPACK as our benchmarks. All of the benchmarks were compiled into ARM statically linked binaries with GNU GCC version 4.4.6 and linked with uClibc library 0.9.30.2. The version of QEMU we used is 2.1.50. MC2LLVM is layered on top of LLVM version 3.2.

We compiled SPEC CPU2006 with two configurations: Option A (-O3 -march=armv5 -mfloat-abi=softfp -mfpu=neon -ftree-vectorize) to tell the compiler to try to generate Neon and VFP instructions, and Option B (-O3 -march=armv5 -mfloat-abi=soft) to tell the compiler not to generate Neon and VFP instructions. The LINPACK benchmark was compiled with -O3 -march=armv5 -mfloat-abi=softfp -mfpu=vfpv3.

Among 12 integer benchmarks (CINT2006) in SPEC CPU2006, perlbench cannot be compiled and both MC2LLVM and QEMU failed to run gcc, a benchmark in CINT2006. Among 17 floating-point benchmarks (CFP2006) in SPEC CPU2006, both MC2LLVM and QEMU failed to run gromacs, cactusADM, and sphinx3, and only MC2LLVM failed to run tonto and games (some bugs happened when emulating them even with Option B, without Neon and VFP support).

B. Performance

Figure 8 (a) and (b) show our comparative approaches. Figure 9 shows the execution time ratio of SPEC CINT2006 compiled with Option A and Option B running on MC2LLVM and QEMU, respectively. The geometric means of the execution time ratio for MC2LLVM and QEMU are 1.174 and 1.052 respectively. The results imply that MC2LLVM could process the Neon and VFP instructions more effectively than QEMU on average.

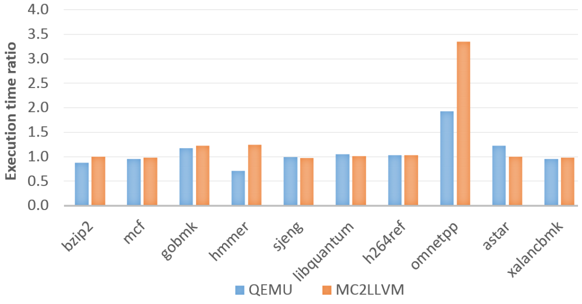


Figure 9. Execution time ratio of Option B/Option A on translating SPEC CINT2006.

VI. Conclusion

We have finished the Neon and VFP extensions in our binary translator MC2LLVM:

1) 1240 translations of the Neon and VFPv3 instructions are emulated.
2) Emulation of the machine features of Neon and VFPv3 architectures is included.

We enhance the translation capability and increase opportunities of using host SIMD and floating-point units to improve performance in MC2LLVM. We also propose new methods by diagnosing the input and output of a floating-point operation to detect floating-point exceptions if the IR layer of a binary translation system do not provide the relevant information about their occurrences instead of emulating a floating-point operation in software, which takes more processor time. We also developed a verification framework for testing the emulated instructions in Neon and VFP to gain confidence in the correctness of our approaches. The experiment results indicate that MC2LLVM is, in average, 1.24X and 2.27X faster than QEMU on the SPEC CPU2006 integer benchmarks and floating-point benchmarks, respectively, and have 3.36X more throughput of the floating-point operations than QEMU benchmarked by LINPACK.

Acknowledgment

The work reported in this paper is partially supported by National Science Council (NSC), Taiwan, Republic of China, under grants MOST 103-2221-E-009-105-MY3, NSC 100-2218-E-009-009-MY3, and NSC 100-2218-E-009-010-MY3.

References

[1] ARM Limited, ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Errata Markup, ARM DDI 0406B, 2011.

[2] V. Bala, E. Duesterwald, and S. Banerjia, “Dynamo: a Transparent Dynamic Optimization System,” ACM SIGPLAN Notices, 35, 5, pp. 1-12, 2000.

[3] F. Bellard, “QEMU, a Fast and Portable Dynamic Translator,” In *Proc. 2005 USENIX Annual Technical Conf.*, pp. 41-46, 2005.

[4] J.-Y. Chen, W. Yang, T.-H. Hung, H.-M. Su, and W.-C. Hsu, “A Static Binary Translator for Efficient Migration of ARM based Applications,” In Proc. 6th Workshop on Optimizations for DSP and Embedded Systems, 2008.

[5] A. Chernoff, M. Herdeg, R. Hookway, C. Reeve, N. Rubin, T. Tye, S.B. Yadavalli, and J. Yates, “FX!32 - A Profile-Directed Binary Translator,” IEEE Micro 18, 2, pp. 56-64, 1998.

[6] K. Hazelwood, G. Lueck, and R. Cohn, “Scalable Support for Multithreaded Applications on Dynamic Binary Instrumentation Systems, In *Proc 2009 International Symp. Memory management* (Dublin, June 19-20, 2009), 2009.

[7] J.L. Henning, “SPEC CPU2006 Benchmark Descriptions,” ACM SIGARCH Computer Architecture News, 34, 4, pp. 1-17, 2006.

[8] C.A. Lattner, LLVM: An Infrastructure for Multi-Stage Optimization, Master’s Thesis, Comp. Sci. Dept, Univ. Illinois at Urbana-Champaign, 2002.

[9] W. Kahan, IEEE Standard 754 for Binary Floating-Point Arithmetic, Lecture Notes on the Status of IEEE 754, pp. 94720-1776, 1996.

[10] MCDisassembler, http://llvm.org/docs/doxygen/html/classllvm_1_1MCDisassembler.html.

[11] R.W. Moore, J.A. Baiocchi, B.R. Childers, J.W. Davidson, and J.D. Hiser, “Addressing the Challenges of DBT for the ARM Architecture,” In Proc. 2009 ACM SIGPLAN/SIGBED Conf Languages, Compilers, and Tools for Embedded Systems (LCTES 09), pp. 147-156, 2009.

[12] B.Y. Shen, J.Y. You, W. Yang, and W.-C. Hsu, “An LLVM-based Hybrid Binary Translation System,” In *Proc. 7th IEEE International Symp. Indurstrial Embedded System* (SIES 12, Karlsruhe, Germany, June 20-22), 2012.

[13] B.-Y. Shen, J.-Y. Chen, W.-C. Hsu, and W. Yang. 2012. “LLBT: an LLVM-based Static Binary Translator,” In Proc. 2012 International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES’ 12), pp. 5160, October 2012.

[14] R.L. Sites, A. Chernoff, M.B. Kirk, M.P. Marks, and S.G. Robinson, “Binary Translation,” Communications of the ACM, 36, 2, pp. 6981, February 1993.

[15] J.E. Smith and R. Nair. Virtual Machines: Versatile Platforms for Systems and Processes. Morgan Kaufmann, June 2005.

[16] Texas Instruments. Pandaboard. OMAP4430 SoC dev. board, revision A2, 2012.

[17] VMware Inc, VMware Workstation, 2013.

[18] C. Zheng and C. Thompson, “PA-RISC to IA-64: Transparent Execution, No Recompilation,” Computer 33, 3, pp. 47-52, March 2000.

Secure Information and Services Management in the Cloud

Marek R. Ogiela, Lidia Ogiela, Urszula Ogiela
AGH University of Science and Technology
Cryptography and Cognitive Informatics Research Group
Krakow, Poland
e-mail: {mogiela, logiela, ogiela}@agh.edu.pl

***Abstract*—In this paper, a new methodology of linguistic threshold schemes application for secure information distribution and management in Cloud Computing will be presented. Linguistic threshold schemes were proposed as extension protocols dedicated for secure information splitting and hierarchical management in different information structures. Currently, Cloud Computing infrastructure offers different resources and services virtualization and distribution, so one of the promising solutions is application of linguistic schemes for splitting of classified information or remote application execution, based on personal accessing grants. The essence of the presented approach is the application of a personally defined formal language, which allows splitting and securely managing strategic data in Cloud Computing. An example of the application of linguistic threshold procedure will also be presented.**

Keywords—cryptographic protocols; secret sharing algorithms; Cloud Computing.

I. INTRODUCTION

One of the emerging problems in data and services security is high level confidentiality and secrecy management in Cloud Computing infrastructure. Security management tasks related to distributed data shares have been intensively developed for various communication infrastructures [3][4][14][16][18]. In this contribution, we will describe the most important aspect related to the use of linguistic models for secrecy management thanks to the creation of a new secure information management protocol. Such new protocol will be based on mathematical linguistic techniques applied for information sharing and encoding, and called linguistic threshold schemes [13].

Mathematical linguistic formalisms have been proposed for computer modeling of natural languages, but later, in addition to such applications, other important areas of application appeared. All possible areas of application of such techniques cover:

- Natural languages description and modeling;
- Compiler construction;
- Pattern classification [6];
- Cognitive analysis [7][8].

Recently, a new field for applying such techniques appeared, and was connected with advanced algorithms for information sharing [13][14][15]. The paper is organized as follows: Section 2 presents the general idea of using linguistic formalisms in creating threshold schemes. In Section 3, we will present an illustrative example showing how information may be divided using such procedures and distributed in the cloud. In Section 4, some concluding remarks will be presented.

II. LINGUISTIC APPROACH FOR DATA AND SERVICES MANAGEMENT

Secure data and services distribution and management are very challenging tasks. For this purpose, we will consider the distributed approach, in which each node will execute different information parts or store different service resources. To distribute information in a hierarchical manner, linguistic schemes may be applied [1][2][8][9].

The general methodology of such protocol is as follows:

- Data sharing schemes should be selected for a particular infrastructure or information management in the cloud;
- The secret data or strategic information should be converted to the bit notation;
- A formal grammar should be introduced to define a new linguistic representation;
- The linguistic representation is divided using the threshold procedure;
- Secret parts may be distributed among different instances in the cloud (Figure 1).

In such a procedure, the divided data or service information [7] distributed over the cloud infrastructure may have a different representation, e.g. in the form of bit blocks with various length or in the form of small images, text sequences, etc.

A procedure which allows generating secret parts of any shared information may be realized in the following way:

- Select appropriate classes of linguistic schemes used for considered data sharing [11];
- Create a linguistic representation of shared data [10];

- Define a grammar generating data shadows (secret parts);
- Distribute generated parts of secret data between different instances.

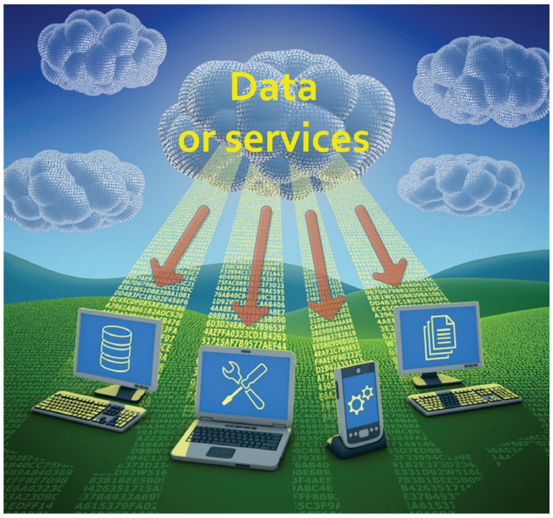


Figure 1. Data or services distribution in Cloud Computing.

The most important feature of the presented protocol is the possibility to use some personal information or encoding keys for the divided information representation during the encoding stage. This means that we can rely on using unique digital sequences or approaches for encoding particular bits or bit blocks of shared information [12][13].

A generalized grammar capable of converting blocks of information to a new representation, which constitutes the shared secret at subsequent stages, can be found in [14].

An introduction for encoding grammar makes it quicker to re-code the input representation of the shared information, which will then be split among different authorized instances. An example of generating data shares using this approach is presented in Figure 2.

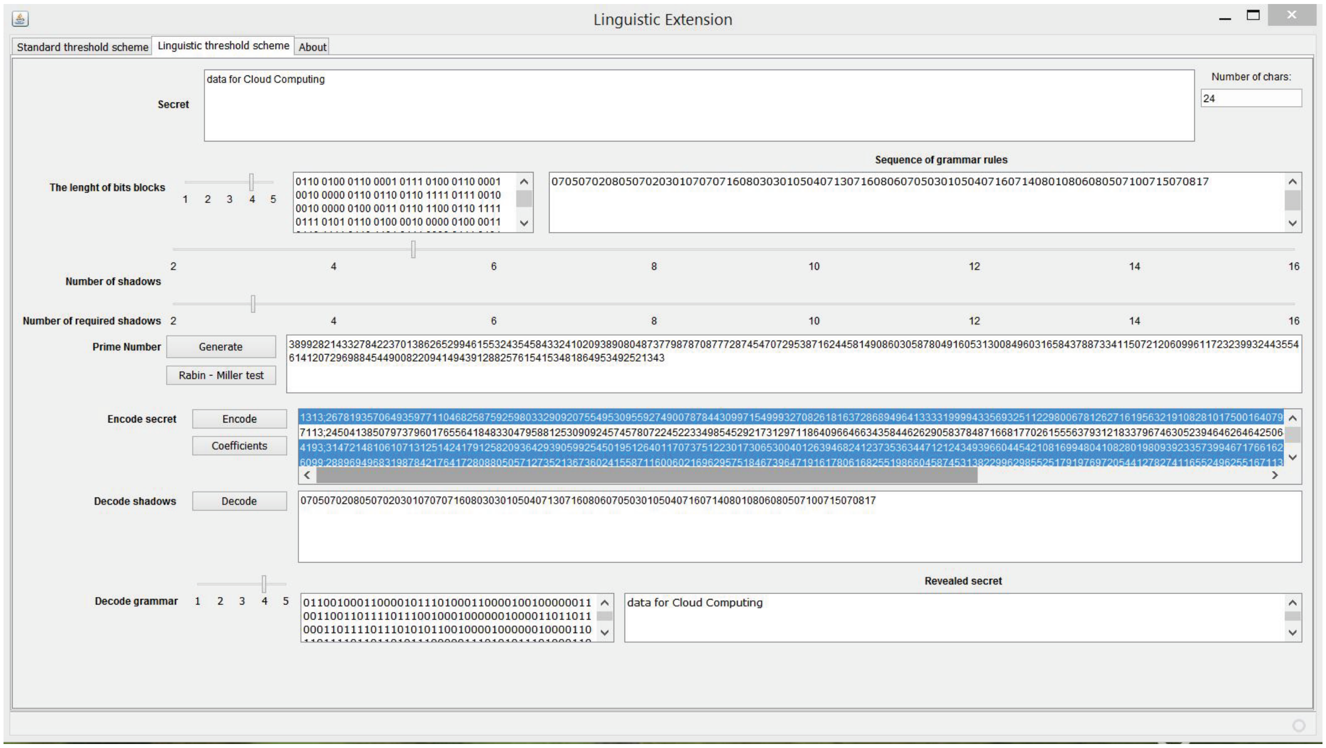
Such a procedure of information splitting in the Cloud infrastructure allows distributing data which are not available to all instances to be securely delivered to trusted instances. The security features of secret splitting algorithms are due to using cryptographic information encryption at the stage of developing these secret parts. The specialized protocols guarantee the security of the entire splitting process, namely, information encryption, its splitting, and later, its reconstruction [13].

III. INFORMATION DISTRIBUTION IN THE CLOUD

In this section, an example of encrypting and reconstructing secret information will be presented. The secret data will have the text form containing the title of this paper, i.e., “Secure Information and Services Management in the Cloud”.

This information can be shared in the way presented in Figure 3, by using a linguistic threshold procedure.

This information may also be presented in a binary form, with 5-bit long blocks marked. Bit blocks of this length will then be coded using a defined grammar.



Secret: data for Cloud Computing

Number of chars: 24

The lenght of bits blocks: 4

```
0110 0100 0110 0001 0111 0100 0110 0001
0010 0000 0110 0110 0110 1111 0111 0010
0010 0000 0100 0011 0110 1100 0110 1111
0111 0101 0110 0100 0010 0000 0100 0011
```

Sequence of grammar rules:

```
07050702080507020301070707160803030105040713071608060705030105040716071408010806080507100715070817
```

Number of shadows: 5

Number of required shadows: 3

Prime Number:

```
3899282143327842237013862652994615532435458433241020938908048737798787087772874547072953871624458149086030587804916053130084960316584378873341150721206099611723239932443554
614120729698845449008220941494391288257615415348186495349252 1343
```

Encode secret:

```
1313;26781935706493597711046825875925980332909207554953095592749007878443099715499932708261816372868949641333319999433569325112298006781262716195632191082810175001640 79
7113;2450413850797379601765564184833047958812530909245745780722452233498545292173129711864096646634358446262905837848716681770261555637931218337967463052394646264642506
4193;3147214810610713125142417912582093642939059925450195126401170737512230173065300401263946824123735363447121243493966044542108169948041082801980939233573994671766162
6099;2889694968319878421764172808805057127352136736024155871160060216962957518467396471916178061682551986604587453138229962985525179197697205441278274116552496255167113
```

Decode shadows:

```
07050702080507020301070707160803030105040713071608060705030105040716071408010806080507100715070817
```

Decode grammar: 4

```
01100100011000010111010001100001001000000011
00110011011110111001000100000001000011011011
00011011110111010101100100001000000010000110
```

Revealed secret: data for Cloud Computing

Figure 2. An example of application of linguistic threshold procedure for shadow generation.
The length of bit blocks is equal to 4, and the shares are generated using a 3, 5-threshold approach.

Thus, the bit representation of the secret takes the following form (Figure 3(A)):

01010 01101 10010 10110 00110 11101 01011 10010 01100 10100 10000 00100 10010 11011 10011 00110 01101 11101 11001 00110 11010 11000 01011 10100 01101 00101 10111 10110 11100 01000 00011 00001 01101 11001 10010 00010 00000 10100 11011 00101 01110 01001 11011 00110 10010 11000 11011 00101 01110 01100 10000 00100 11010 11000 01011 01110 01100 00101 10011 10110 01010 11011 01011 00101 01101 11001 11010 00010 00000 11010 01011 01110 00100 00001 11010 00110 10000 11001 01001 00000 01000 01101 10110 00110 11110 11101 01011 00100

In order to encode such a digital sequence, a special context-free grammar should be defined, which allows to encode 5-bit block using terminal symbols [13].

From the security point of view, the grammar and the length of terminal symbols remain secret, but known only to the instances that generate and distribute secret parts. For the obtained bit representation of the secret, and the defined grammar, a syntactic analysis should be performed, which will generate a new sequence containing the numbers of grammar rules. Such grammar rules allow generating the bit sequence using terminal and non-terminal symbols. This sequence of production numbers has the following form (Figure 3(B)). For our secret information, the sequence of grammar rules has the following form:

1114192307301219132117051928200714302607272512211406242329090402142619030121280615102807192528061513170527251215130620231128120614262703012712150502270717261001091423073130120533

Such new representation of the secret in the form of a sequence of production numbers is the basis for further steps of information sharing using the selected threshold algorithm. For this, a large prime number should be generated (Figure 3(C)).

Then, we use the sharing algorithm to share the secret using a (3, 6) threshold scheme. This means that 6 shares of the secret are generated, but to reconstruct it again it will be necessary to combine at least 3 freely selected shares (Figure 3(D)). For our example, the generated shadows are presented in Figure 3(E).

The trusted instance which executes the secret procedure can distribute the obtained shares among authorized persons or instances.

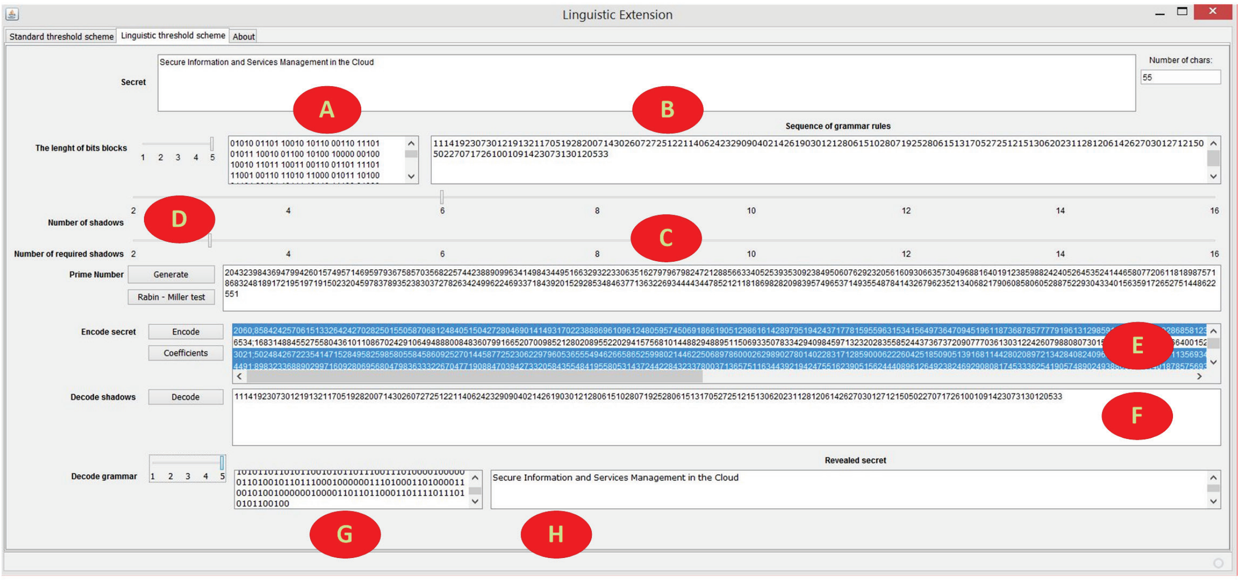
The reconstruction of the original information distributed between the authorized instances may be realized in the way described below.

At the beginning, the selection of the necessary number of secret shares must be performed. In our case, we collect at least three shadows (Figure 3(E)), which allow reconstructing the original input sequence, which is really the sequence of production rules presented in Figure 3(F).

In the next step, the rules of the defined grammar should be applied to the sequence obtained above and the production numbers should be replaced with bit sequences represented by these rules. This produces the previous binary representation of the originally shared secret information.

The instance, which is necessary to restore the secret, uses their knowledge of the grammar employed to encrypt the information and converts the sequence of production numbers obtained in the previous stage into a binary representation of the secret presented in Figure 3(G).

At the end of the secret reconstruction process, the managing instance reconstructs the final secret information having the form of letters, digits or other characters using information about the coding method in the given system. This allows the original secret to be reconstructed as presented in Figure 3(H).



Secret: Secure Information and Services Management in the Cloud

Number of chars: 55

The lenght of bits blocks: 5

```
01010 01101 10010 10110 00110 11101
01011 10010 01100 10100 10000 00100
10010 11011 10011 00110 01101 11101
11001 00110 11010 11000 01011 10100
```

Sequence of grammar rules

```
1114192307301219132117051928200714302607272512211406242329090402142619030121280615102807192528061513170527251215130620231128120614262703012712150
502270717261001091423073130120533
```

Number of shadows: 6

Number of required shadows: 3

Prime Number

```
2043239843694799426015749571469597936758570356822574423889099634149843449516632932233063516279796798247212885663340525393530923849506076292320561609306635730496881640191238598824240526453524144658077206118189875718
68324818917219519719150232045978378935238303727826342499622469337184392015292853484637713632269344443447852121181869828209839574965371493554878414326796235213406821790608580605288752293043340156359172652751448622
551
```

Encode secret

```
2060;858424257061513326424270282501550587068124840515042728046901414931702238886961096124805957450691866190512986161428979519424371778159559631534156497364709451961187368785777791961312985928685812
6534;1683148845527558043610110867024291064948880084836079916652070098521280208955220294157568101448829488951150693350783342940984597132320283558524437367372090777036130312242607988080730156400152
3021;502484267223541471528495825985805584586092527014458772523062297960536555494626658652599802144622506897860002629890278014022831712859000622260425185090513916811442802089721342840824096113569
4491;898323368890299716092806956804798363332267047719088470394273320584355484195580531437244228432337800371365751163443921942475516239051562444089612649238246929080817453336254190574890249381878575693
```

Decode shadows

```
1114192307301219132117051928200714302607272512211406242329090402142619030121280615102807192528061513170527251215130620231128120614262703012712150502270717261001091423073130120533
```

Decode grammar: 5

```
10101110110101100101011011100111010000100000
0110100101101110001000000111010001101000011
0010100100000010000110110110001101111011101
0101100100
```

Revealed secret: Secure Information and Services Management in the Cloud

Figure 3. Stages of secret sharing having textual form.

At the same time, it is worth noting that if the required number of shares of the divided information necessary to restore it is not selected when restoring the secret, the use of the threshold scheme will not generate the sequence of production numbers of our grammar, but some nonsensical information that cannot be converted into a meaningful text.

The linguistic threshold procedure is very universal and may have many different applications. The main application is related to intelligent secret data distribution in hierarchical management structures [6][16][18]. In such management models, at each level in the management pyramid, particular secret data may be divided in different manners depending on the level, number of trusted persons and accessing grants to secret information.

Such protocols may also play an important role in secret data distribution in cloud infrastructures. In such case, storing different secret parts on different cloud servers will affect for information confidentiality and security of transmission over the distributed networks. Such features are especially important in storing and managing a big data or large information repositories. The efficiency of linguistic threshold procedures was evaluated in [13].

IV. Conclusion and Future Work

We presented a new protocol for digital information sharing and distribution in cloud infrastructure. In particular, we proposed a new scheme for secret division and distribution between trusted instances, based on linguistic threshold procedures. The secret parts of the divided information may be obtained using the linguistic threshold procedure presented in this paper, which allows sharing data using formal grammars in a more general way than in DNA cryptography [11]. Such a method of information encoding and sharing may define new areas in cryptography and security called cognitive cryptography. The presented approach seems to be very useful in efficient information management in Cloud infrastructures [15][17].

The main difference between the presented linguistic approach and classic encoding cryptographic procedures is that our protocol is a high level sharing algorithm, allowing the generation of secret parts using both traditional threshold procedures as well as specially defined (by splitter or user) grammar. This grammar allows controlling the sharing stage in the manner defined by the splitter. The whole protocol is secure and efficient in generating secret parts of information [3][4][5]. The grammar may also influence the security level in this protocol. If the grammar is more complex the cryptanalysis will be more difficult because it will be necessary to determine the larger number of applied grammar rules.

The presented approach has also a great number of possible application, especially in solving practical security related and management problems. In particular, it is dedicated to dividing secret or classified information and distribute it in a secure manner between trusted parties. It guarantees not only security and confidentiality of information, but also prevents against insider treats or unauthorized access for strategic data stored in Cloud Computing infrastructure or distributed architectures.

Acknowledgment

This work has been supported by the National Science Centre, Republic of Poland, under project number DEC-2013/09/B/HS4/00501.

References

[1] G. Ateniese, C. Blundo, A. de Santis, and D.R. Stinson, “Visual cryptography for general access structures,” Information and Computation, vol. 129, 1996, pp. 86–106.

[2] A. Beimel, and B. Chor, “Universally ideal secret sharing schemes,” IEEE Transactions on Information Theory, vol. 40, 1994, pp. 786–794.

[3] S. Haag, and M. Cummings, “Management Information Systems for the Information Age,” McGraw-Hill, Irwin, 2012.

[4] N. Hidayah Ab Rahman, and K-K. R. Choo, “A survey of information security incident handling in the cloud,” Computers & Security, vol. 49, 2015, pp. 45–69

[5] P. Li (et al.), “Essential secret image sharing scheme with different importance of shadows,” J. of Visual Communication and Image Representation, vol. 24, 2013, pp. 1106–1114.

[6] O.J. Mackenzie, Eds. “Information Science and Knowledge Management,” Springer-Verlag, Berlin, 2006.

[7] A. Menezes, P. van Oorschot, and S. Vanstone, “Handbook of Applied Cryptography,” CRC Press, Waterloo, 2001.

[8] M.R. Ogiela, and U. Ogiela, “Linguistic Extension for Secret Sharing (m, n)-threshold Schemes,” SecTech 2008 – 2008 International Conference on Security Technology, December 13-15, 2008, Hainan Island, Sanya, China, pp. 125–128.

[9] M.R. Ogiela, and U. Ogiela, “Shadow Generation Protocol in Linguistic Threshold Schemes,” Communications in Computer and Information Science, vol. 58, 2009, pp. 35-42.

[10] M.R. Ogiela, and U. Ogiela, “Security of Linguistic Threshold Schemes in Multimedia Systems,” Studies in Computational Intelligence, vol. 226, Springer-Verlag, Berlin Heidelberg, 2009, pp. 13–20.

[11] M.R. Ogiela, and U. Ogiela, “Grammar Encoding in DNA-Like Secret Sharing Infrastructure,” LNCS, vol. 6059, 2010, pp. 175–182.

[12] M.R. Ogiela, and U. Ogiela, “Linguistic Protocols for Secure Information Management and Sharing,” Computers & Mathematics with Applications, vol. 63(2), 2012, pp. 564–572.

[13] M.R. Ogiela, and U. Ogiela, “Secure Information Management using Linguistic Threshold Approach,” Advanced Information and Knowledge Processing, Springer-Verlag London 2014.

[14] N. Pakniat, M. Noroozi, and Z. Eslami, “Secret image sharing scheme with hierarchical threshold access structure,” J. of Visual Communication and Image Representation, vol. 25. 2014, pp. 1093–1101.

[15] J. Rhoton, “Cloud Computing Protected: Security Assessment Handbook,” Recursive Press, 2013

[16] J.R. Schermerhorn, “Management,” Wiley, 2012.

[17] S. Subashini, and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” Journal of Network and Computer Applications, vol. 34, 2011, pp. 1–11.

[18] M.E. Whitman, and. H.J. Mattord, “Management of Information Security,” Stamford, USA, Cengage Learning, 2014.

Hybrid Autonomous Navigation System Using a Dynamic Fuzzy Cognitive Maps Evolution

Márcio Mendonça[+] / Lúcia Valéria R. Arruda[*]
[+]DAELE (Electric Department) UTFPR-CP
Cornélio Procópio, Brazil;
[*]CPGEI, UTFPR-CT;
Curitiba, Brazil;
e-mails: {mendonca; lvrarruda}@utfpr.edu.br

Ivan Rossato Chrun[++]/ Esdras S. da Silva[**]
[++]CPGEI (graduate master degree in Electrical Engineering)
UTFPR-CT
Curitiba, Brazil;
[**]COEME (graduate master degree in Mechanical
Engineering) UTFPR-CP
Cornélio Procópio, Brazil;
e-mails: ivanchrun@gmail.com; esdras.utfpr@bol.com.br

***Abstract*—This work develops a knowledge based system using Fuzzy Cognitive Maps (FCM) for autonomous navigation. A new variant of FCM, named Dynamic-Fuzzy Cognitive Maps (D-FCM), is used to model decision, tasks and/or make inference in the robot/mobile navigation. Fuzzy Cognitive Maps is a tool that models qualitative structured knowledge through concepts and causal relationships. The proposed model allows representing the dynamic behavior of the mobile robot (agent) in different environments. A brief review of correlated works in navigation area using FCM and Fuzzy Systems is presented. Some simulation results are discussed highlighting the ability of the mobile to navigate among obstacles and reach targets (navigation environment).**

Keywords - Fuzzy Cognitive Maps; Autonomous Navigations; Hybrid Architecture; Intelligent dynamic decision systems.

I. INTRODUCTION

Artificial Intelligence (AI) has applications and development in various areas of knowledge, such as mathematical biology, neuroscience, computer science, swarm robotics and others. The research area of intelligent computational systems aims to develop methods that try to mimic or approach the human's capabilities to solve problems. These news methods are looking to emulate human's abilities to cope with very complex processes, based on inaccurate and/or approximated information. However, this information can be obtained from the expert's knowledge and/or operational data or behavior of an industrial system [1].

Researches in autonomous navigation are in stage of ascent. Autonomous Systems have the ability to perform complex tasks with a high degree of success [2]. In this context, the complexity involved in the task of trajectory generation is admittedly high and, in many cases, requires the autonomous system being able to learn a navigation strategy through interaction with the environment [3].

There is a growing interest in the development of autonomous (agents) robots and vehicles, mainly because of the great diversity of tasks that can be carried out by them, especially those that endanger human health and/or the environment, [4][5]. As an example, we can cite Mandow et al. [6], which describe an autonomous mobile robot for use in agriculture, in order to replace the human worker through inhospitable activities, as spraying with insecticides.

The problem of mobile robots control comprises two main sub problems: 1) navigation, determining of robot/vehicle position and orientation at a given time, and 2) guided tours, which refers to the control path to be followed by the robot/vehicle. Some cases have more complexity; e.g., a Hybrid Architecture [7] is used to develop Dynamic Fuzzy Cognitive Maps-based models for autonomous navigation with different goals: avoiding obstacles, exploration, and reach targets.

This work specifically proposes the development of an autonomous navigation controller system using heuristic knowledge about the behavior of the robot/vehicle in different scenarios, modeled by Fuzzy Cognitive Maps (FCM) [8]. In this case, the robot/vehicle determines sequences of action in order to reach a given goal state from a predefined starting state.

The FCM combines aspects of Artificial Neural Networks [2], Fuzzy Logic [1], Semantics Networks [2] and others intelligent systems techniques. Through cognitive maps, beliefs or statements, regarding a limited knowledge domain, are expressed through words or phrases, interconnected by simple relationship of cause and effect (question/non-question). In the proposed model, the FCM relationships are dynamically adapted by rules that are triggered by the occurrence of special events. These events must change mobile behavior. There are several works in the literature that model heuristic knowledge necessary for decision-making in autonomous navigation, e.g., Classic Fuzzy and FCM Systems [9]-[13]. In a similar way, the proposed approach in this paper is to build qualitative models for mobile navigation by means of fuzzy systems. However, the knowledge is structured and built as a cognitive map representing the behavior of the mobile.

Therefore, the proposed autonomous navigation system must be able to take dynamic decisions to move through the environment and change its trajectory as a result of an event. For this, the proposed FCM model must aggregate discrete and continuous knowledge about navigation. Actions, such as the decision to turn left or right, when sensors accuse obstacles, and accelerate, when there is a free path, are always valid control actions in all circumstances. In this way, this type of action is modeled as causal relationship in a classical FCM.

However, there are specific situations, such as the need to maintain a trend of motion, mainly in curves, when the vehicle is turning left and sensors to accuse a new obstacle in the same direction. Due to inertia and physical restrictions, the mobile cannot abruptly change direction; this type of maneuver must be carefully executed. In this context, some specific situations should also be modeled in the map by causal relationships and concepts, but they are valid just as a result of a decision-making task caused by ongoing events. To implement such a strategy, new types of relationships and concepts will be added to the FCM classic model.

This new type of FCM, in which the concepts and relationships are valid as a result of decision, driven by events modeled by rules is called Dynamic-FCM. Specifically, the work of Mendonça et al. [16] presents a type of D-FCM (ED-FCM), which aggregates the occurrence of events and other facilities that makes appropriate this type of cognitive map, for the development of intelligent control and automation in an industrial environment.

Section II presents the concept of FCM and the FCM applied in autonomous navigation. Section III illustrates the D-FCM model utilized. Section IV presents the Hybrid Architecture and a brief background. Section V shows the results, and finally, Section VI presents the conclusion and future works.

II. Fuzzy Cognitive Maps

Cognitive maps were initially proposed by Axelrod [16] to represent words, thoughts, tasks, or other items linked to a central concept and willing radially around this concept. Axelrod developed also a mathematical treatment for these maps, based in graph theory, and operations with matrices. In general, these cognitive maps are "belief structure" of a person or group, allowing to infer or predict the consequences that this organization of ideas represents in the graph; in cognitive maps, a central concept is not necessary.

This mathematical model was adapted for inclusion of Fuzzy logic uncertainty. In specific, linguistics terms generating widespread fuzzy cognitive maps. Like the original, FCMs are directional graphs, in which the numeric values are fuzzy sets or variables. The "graph nodes", associated to linguistic concepts are represented by fuzzy sets and each "node" is linked with others through connections [8]. Each of these connections has a numerical value (weight), which represents a fuzzy variable related to the strength of its concepts.

The concept of a cognitive map can be updated through the iteration with other concepts and with its own value. In this context, a FCM uses a structured knowledge representation through causal relationships being calculated mathematically from matrix operations, unlike much of intelligent systems whose knowledge representation is based on rules if-then type. However, due to this "rigid" knowledge representation the FCM-based inference models lack robustness in presence of dynamic modifications not a priori modeled [17]. To circumvent this problem, this article develops a new type of FCM, in which concepts and causal relationships are dynamically inserted into the graph from the occurrence of events. In this way, the dynamic fuzzy cognitive map model is able to dynamically acquire and use the heuristic knowledge. The proposed D-FCM and its application in autonomous navigation will be developed and tested in the following sections.

Related work using cognitive maps in the robotics research area can be found in the literature. Among them, it can be cited the work by Min et al. [12], that employs probabilistic FCM in the decision-making of a soccer robot team. These actions are related to the behavior of the team, such as kick the ball in presence of opponents. The probabilistic FCM aggregates a likelihood function to update the concepts of the map. A FCM is used by Pipe [13] to guide an autonomous robot. The FCM is designed from a priori knowledge of experts and afterwards it is refined by a genetic algorithm.

The inference process of the FCM model can be calculated by the following rule given in (1) and (2); these equations are used in several works in the literature, e.g., FCM and evolutions such as Mendonça et al. [7] and Siraj, Bridges and Vaughn [9].

$$A_i = f(\sum_{\substack{j=1 \\ j \neq i}}^{n} (A_j \times W_{ji}) + A_i^{old} \tag{1}$$

where:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \tag{2}$$

A review of correlated works in indoor autonomous navigation robotics can be found in [14]. The objective of this paper is to develop an autonomous explorer agent (robot) based in a low cost and open source platform with the ability to tune FCM model by interacting with the environment. The agent architecture is inspired by Braitenberg [15], who suggests the application of computational intelligence techniques, starting up with a simple model with one or only a few functionalities, and gradually adding new objectives to improve the exploration capability of the agent.

However, our navigation system does not use a priori information about the environment. The FCM represent the usual navigation actions as turn right, turn left, accelerate and others. The adaptation ability to environment changes and to take decisions in presence of random events is reached by means of a rule-based system. These rules are triggered in accordance of "intensity" of the sensor measurements. In this research, the kinematic model use sensors signal and pulses in the right and left wheels.

III. THE D-FCM MODEL

The development of a FCM model is similar of the work of Mendonça et al.’s [16]. In this case, we identify 3 inputs related to the description of the environment (presence of obstacles) and 2 outputs describing the mobile’s movements: turn left, turn right or forward (pulses in both wheels). The three inputs take values from the three sensors located at left, right and front side of the mobile.

These concepts are connected by arcs representing the actions of acceleration (positive) and braking (negative). Three decisions are originally modeled, if left sensor accuses an obstacle, the vehicle must turn to the right side and equally if the right sensor accuses an obstacle in the right side, the vehicle turns to the other side. The direction change decision implies in smoothly vehicle deceleration. The third decision is related to a free obstacle environment; in this case, the mobile follows a straight line accelerating smoothly.

Figure 1 shows the robot (agent) for used as kinematic model development; however, it is not in the scope of this work to demonstrate the equations of the model used in the simulations, only the development and simulated results of the proposed controller. The kinematic model used have is similar physic characteristics, e.g., geometry (axes distance) and inputs and outputs development. The input concepts are LS (Left sensor), RS (Right sensor) and FS (Frontal sensor) and the output concepts are LW (Left Wheel Pulse) and RW (Right Wheel Pulse). The values of the concepts are the readings of the corresponding sensors. As a fuzzy number, these values are normalized into the interval [0, 1].

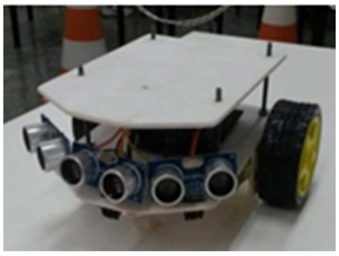


Figure 1. Structure Generic Robot using Arduino Mega

The future navigation prototype has its position estimated by the numbers of pulses given by the step motors and the obstacle avoidance is guided by the navigation system. However, the prototype is under development and the focus of this work is in its initial results obtained by the simulator. Whereas the environment is partially known, only the targets have their position known by the robot (agent). In simulation time, the robot (agent) knows its position, and the new control actions are calculated by the D-FCM, sequentially at every step.

In this work, if the target is located to the left of the robot, DSx is negative and is located at the rear of the robot, DSY is negative. The concept used is "DSx" to the lateral distance between the robot and the target (ΔX), and "DSy" to the front distance (ΔY).

Figure 2 shows the simplest case, where the agent goes directly toward the target (known point in the scenario). In this context, and two objectives were developed for the FCMs (avoid obstacles and reach the targets); the target position is known and the agent will alternate between two FCMs (using a finite state machine) to change its objectives.

Figure 3 shows concepts and relationships for avoiding obstacles. In resume, weights w14 and w35 are positive, otherwise the weights w34 and w15 are negatives. These values are necessaries for avoiding maneuvers. The weights w24 and w25 are connected in the frontal sensors and wheels concepts, and have negative values because when obstacle is near, the robot should decelerate.

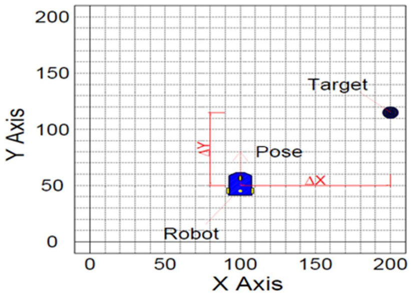


Figure 2. Scenario - Distances

If the target is located to the left of the robot, DSx is negative and is located at the rear of the robot, DSY is negative. The values are dynamically tuned by the Hebb Learning algorithm [17].

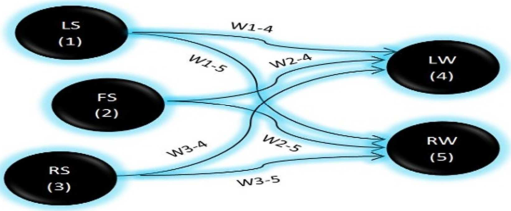


Figure 3. D-FCM avoid obstacles

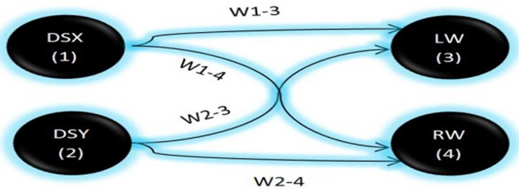


Figure 4. D-FCM reach targets.

Figure 4 shows the D-FCM_2 which its goal is to reach the one or more target using distance previously knowledge (see in Figure 2).

For changing D-FCM_1 and D-FCM_2, a finite state machine is used (Figure 5); the deliberative part of the architecture is discussed in Section IV. The switching is done dynamically according to the occurrence of events, at first the robot will toward the target, however, it changes D-FCM if an obstacle is at a minimum distance of 15cm.

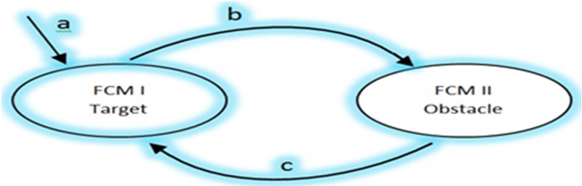


Figure 5. Finite State Machine

The language of the finite state machine is:

- a: Start Machine for reach targets;
- b: Change FCM for avoid obstacles;
- c: Return Machine for reach targets again;

Other developments in the FCM are known in the literature as E-FCM (Extended-FCM) [18], RB-FCM (Rule Based-FCM) [19] and the DCN (Dynamic Cognitive Networks) [20]. A recent survey with major variations of classic FCMs, in recent years, suggesting low computational complexity, is presented by Papageorgiou and Salmeron [21].

IV. Hybrid Architecture Background

Hybrid Architectures aims to combine the main features of deliberative and reactive approaches, trying to reduce the restriction on the scope of each of these approaches. That is, the hybrid architectures use determination to plan the actions of the robot from an internal global representation of the world knowledge, so the objectives of the robot can be achieved efficiently. Once the actions are planned, the action plan implementation is done using reactive interactions between sensors and output system, allowing quick actions towards changes in the environment. These architectures, also use deliberation to plan the actions of the robot from an internal knowledge representation of the world, so the goals of the robot can be achieved efficiently [22][23]. This D-FCM hybrid architecture is also inspired by behavior [24][25].

As shown in Figure 6, the proposed architecture is presented in a generic form to assist the D-FCM development. Each block represents a specific part of the system; the Perception System symbolizes the sensors; the Internal State System, the finite state machine; the Behavior System, the FCM´s; the Learning System, the dynamic learning algorithm; the Motor system, the system output; and at last, the Environment, the interaction with the environment (perception, planning and actions). This means that planning is not part of the perception-action cycle, interacting only when that organization have any relevant result (as an event, e.g., detecting an obstacle) of their planning.

The states modeled in this study are two: get the target, located at a previously known point, and avoid obstacles, without prior knowledge of their position by the perception of sensors.

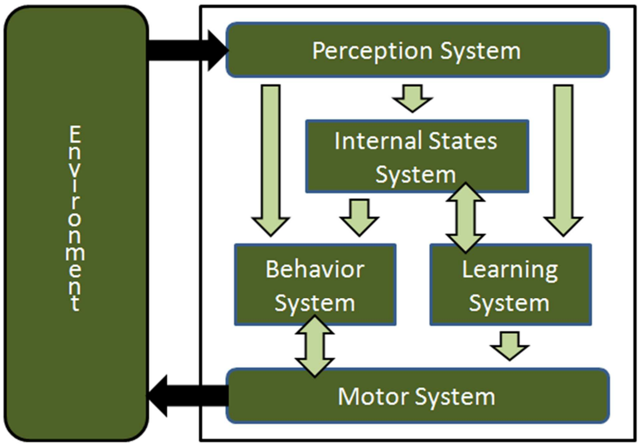


Figure 6. hybrid architecture D-FCM

The motor system is responsible for the agent movement inferences on the environment, according to its current state dynamically tuned by a Hebb learning algorithm [17], through data provided by the perception system; however, it could be also used reinforcement learning algorithms [16]. This generic Hebb equation is:

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \qquad (3)$$

where: η is a positive constant that determines the learning rate, $\boldsymbol{x_j}$ represents the presynaptic signal $\boldsymbol{Y_k}$ represents the postsynaptic signal, $\Delta\boldsymbol{W_{kj}}$ is the synaptic weight $\boldsymbol{n}$. Each of the causal relationships of FCMs (Figures 3 and 4) uses the above equation to dynamically tune theirs weights.

The basic difference between D-FCM and classic FCM is the dynamic tuning ability of causal relationships and switching of two or more structures by state machine, according to the desired goal.

V. Results

A two dimensional simulator was implemented in Matlab to study the dynamic behavior of the mobile agent. Several studies present FCMs results, using simulation, can be found in the literature [7] [12] [13] [17] [18] [20].

The scale used for the simulated scenario is 1:100. In this context, Russel and Norvig [2] suggest that, in order to consider an autonomous agent, it is necessary to succeed in at least three different simulations. Thus, the simulations were tested with different scenarios settings to suggest autonomy. Figures 7-11 show the proposed work simulations; the first two simulations only reach targets, in different scenarios. Each simulation has a crescent level of difficulty, as proposed and specified in [15]; observe that all the scenarios are static.

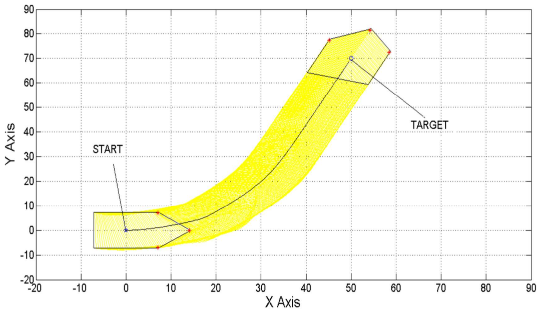


Figure 7. Simulation 1 (Target 1)

The simulation in Figure 7 is simple and shows the trajectory of the agent (robot) toward the target at a specific point, between first and second quadrant. It shows, in yellow, the trail (agent´s pose memory), and finally, it shows the initial and final pose of the agent. This explanation applies to all following figures.

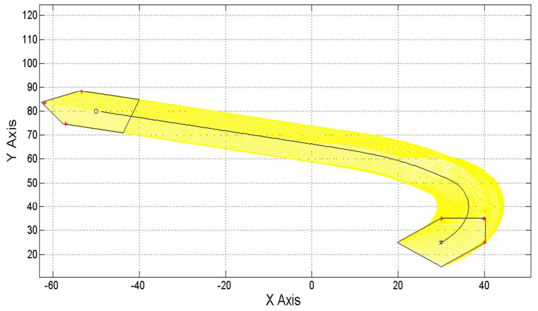


Figure 8. Simulation 2 (target 2)

The second experiment (Figure 8) is similar to the first put the show initial poses and different finals, including in different quadrants. It shows the navigation versatility of the FCM controller.

The experiment in Figure 9 and 10 has an increase in its difficulty, by adding obstacles in the environment. One of the classic challenges is the problem of series decision making, i.e., an error in the second step can have influence in the third one, and so on [16].

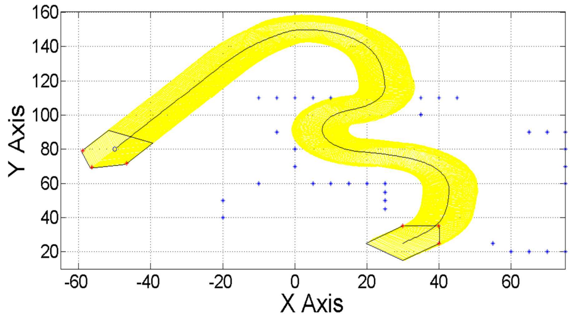


Figure 9. Simulation 3 (target and obstacles)

Figure 9 suggest autonomy due results showing positive outcomes in different scenarios, as already mentioned.

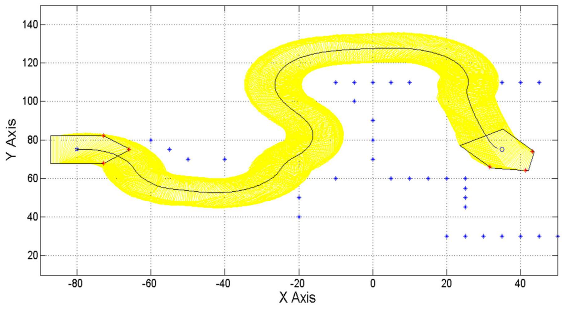


Figure 10. Simulation 4 (target and obstacles)

The simulation in Figure 11 suggests an increase in the autonomy of the controller’s capacity; in particular, it shows the difficulty of reaching the target, in the center of a spiral of obstacles, successfully.

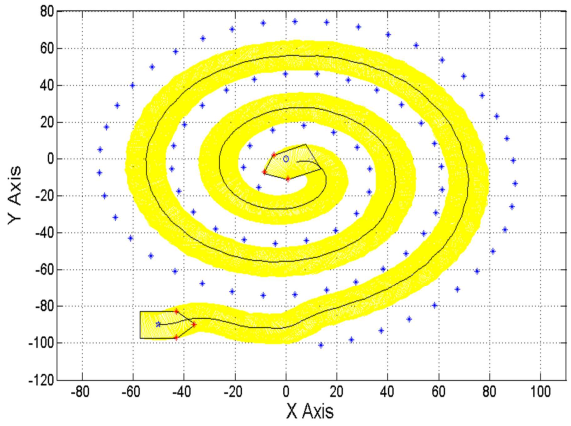


Figure 11. Simulation 5 (spiral)

VI. CONCLUSION

This paper developed a hybrid autonomous navigation architecture based on a new type of fuzzy cognitive maps, named dynamic fuzzy cognitive map, D-FCM.

The initial results obtained from the simulations were convincing, because the mobile agent accomplished the goal of reaching the target with a maximum error of few centimeters deviating from obstacles. However, in a real environment, it is difficult to reach the same precision.

Some difficulties presents in real robots, e.g., ghost signal (in particular, ultrasound sensors), noise, and others, were not considered in the simulations. However, the variations of scenarios with obstacles, highlighting the scenario with a spire of obstacles, suggest that hybrid architectures for autonomous robots navigation can handle achieving goals in different scenarios, with crescent degrees of difficulties.

Future work aims to compare the proposed controller with other intelligent techniques, like Classic Fuzzy or Adaptive Fuzzy and ANFIS, by comparing number of maneuvers and time required for achieving the objectives, and then, by improving the complexity of the scenarios using walls with 90 degrees.

Another target is to embed this system into a real robot using an open source development platform, such as low cost microcontroller (e.g., Arduino), due FCMs low computational complexity. Finally, a test phase is scheduled for the proposed controller in dynamic scenarios, such as in the presence of mobile and/or surprise obstacles.

Acknowledgment

This research is part of the development of post-doctorate from the first author, in UTFPR-Curitiba.

References

[1] M. K. Passino, and S. Yourkovich. Fuzzy control. Menlo Park: Addison-Wesley. 1997.

[2] S. J. Russell, and P. Norvig, Artificial intelligence: a modern approach. Englewood Cliffs: Prentice Hall. 1995.

[3] R. Calvo. Intelligent hybrid architecture for autonomous navigation of robots. Dissertation (Master in Computer Science and Computational Mathematics). IMC-USP. 2007

[4] M.H. Shaikh, K. Kosuri, N.A. Ansari, and M.J. Khan. The state-of-the-art intelligent navigational system for monitoring in mobile autonomous robot. Information and Communication Technology (ICoICT), 2013 International Conference of, 2013, pp.405–409.

[5] T. Maki, A. Kume, T. Ura, T. Sakamaki, and H. Suzuki. Autonomous detection and volume determination of tubeworm colonies from underwater robotic surveys. OCEANS 2010 IEEE - Sydney, 24–27 May 2010, pp.1–8,

[6] A. Broggi, A. Zelinsky, M. Parent, and C. E. Thorpe. Intelligent vehicles. Springer Handbook of Robotics, pp. 1175–1198, 2008.

[7] M. Mendonça, B. A. Angélico, L. V. R. Arruda, and F. Jr. Neves. A Subsumption Architecture to Develop Dynamic Cognitive Network-Based Models with Autonomous Navigation Application. Journal of Control, Automation and Electrical Systems, vol. 1, pp. 3–14, 2013.

[8] B. Kosko. Fuzzy Cognitive Maps. Int.J. Man-Machine Studies, 1986, vol. 24, pp. 65–75.

[9] A. Siraj, S. Bridges, and R. Vaughn. “Fuzzy Cognitive Maps for Decision Support in an Intelligent Intrusion Detection System”, Technical Report, Department of Computer Science, Mississippi State University. MS 39762, 2001.

[10] J. Yang, and N. Zheng. An expert fuzzy controller for vehicle lateral control. 33rd Annual Conference of the IEEE on Industrial Electronics Society (IECON), 2007, pp. 880–885.

[11] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, and L. T. Aguilar. Intelligent Control of an Autonomous Mobile Robot using Type-2 Fuzzy Logic. Engineering Letters, vol. 13, n. 2, 2006.

[12] H.Q. Min, J.X. Hui, Y-S. Lu, and Jz. Jiang. Probability Fuzzy Cognitive Map for Decision-making in Soccer Robotics. Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06) 0-7695-2748-5/06, 2006.

[13] A. G. Pipe. An Architecture for Building “Potential Field” Cognitive Maps in Mobile Robot Navigation. Adaptive Behavior, vol. 8, n. 2, pp. 173–203, 2000.

[14] Di Paola, D. Naso, D. Milella, A. Cicirelli, and G. Distante, Multi-Sensor Surveillance of Indoor Environments by an Autonomous Mobile Robot. Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference on , 2–4 Dec. 2008, pp.23–28.

[15] V. Braitenberg. Vehicles: Experiments in Synthetic Psychology. The MIT Press, 1986.

[16] M. Mendonça, L. V. R. Arruda, and F. A. Neves. Autonomous Navigation System Using Event Driven-Fuzzy Cognitive Maps. Applied Intelligence (Boston), vol. 37, pp. 175–188, 2011.

[17] E. I. Papageorgiou, Learning Algorithms for Fuzzy Cognitive Maps. IEEE Transactions ON Systems and Cybernetics. Part C: Applications and Reviews. vol. 42, pp. 150–163, March. 2012.

[18] M. Hagiwara. Extended *Fuzzy* cognitive maps. In: Proceedings of IEEE international conference on *fuzzy* system, New York, 1992. pp. 795–801.

[19] J. P. Carvalho, and J. A. Tomé. Rule based Fuzzy cognitive maps-qualitative systems dynamics. In: Proceedings 19th international conference of the North America. Fuzzy information fuzzy processing society, 2000, pp. 407–411.

[20] Y. Miao, C. Y. Miao, X. Tao, Z. Shen, and Z. Liu. Transformation of cognitive maps. IEEE Transactions on *Fuzzy* Systems, vol. 18, n. 1, pp. 114–124, Feb. 2010.

[21] E.I. Papageorgiou, and J.L. Salmeron. A Review of Fuzzy Cognitive Maps Research During the Last Decade. Fuzzy Systems, IEEE Transactions on, vol.21, n.1, pp.66–79, Feb. 2013.

[22] L. Wang. Analysis and design of hierarchical fuzzy systems, IEEE Trans. Fuzzy Syst., vol. 7, pp. 617–624,1999.

[23] R. A. F. Romero. Robótica Móvel. LTC, 07/2014. VitalBook file. [portuguese e-book].

[24] R. A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, Mar. 1986. vol. 2, n. 1, pp. 14–23.

[25] C.A. Policastro, R.A.F. Romero, and G. Zuliani. Robotic architecture inspired on behavior analysis. Neural Networks, 2007. IJCNN 2007. International Joint Conference on, 12–17 Aug. 2007, pp.1482–1487.

Effective Interaction in Asynchronous Multi-Agent Environments for Supply Scheduling in Real-Time

Alexander Tsarev
Knowledge Genesis Group Ltd.
Samara, Russia
email: mail@identifier.at

Petr Skobelev
Samara State Aerospace University
Samara, Russia
email: petr.skobelev@gmail.com

Dmitry Ochkov
Smart Solutions Ltd.
Samara, Russia
email: ochkov@smartsolutions-123.ru

***Abstract*—This paper focuses on analysis of effective interaction techniques of agents in multi-agent systems used for real-time scheduling. The paper describes two approaches to the organization of the interaction of asynchronously working software agents. The supply network scheduling case is considered to show the difference in how the interaction goes on. The comparison shows how well each approach allows parallel processing, and subsequently, how fast the scheduling can be done on multi-core hardware. The pros and cons of the approaches are described, as well as ways to achieve better quality. Finally, the results of processing of real data using the approaches are given. The results show a higher effectiveness of one of the approaches in real-time supply scheduling.**

***Keywords*—real-time; scheduling; software agent; multi-agent; supply chain; supply network; supply demand; interaction protocol; agent negotiation; asynchronous interaction; processing speed; parallel processing; schedule quality.**

I. Introduction

Growing complexity and dynamics of modern global market demand new paradigms in resource management [1][2]. New revolutionary approach to increase efficiency of business is associated today with real-time economy, which requires adaptive reaction to events, ongoing decision making on resource scheduling and optimization and communication results with decision makers.

Multi-agent technology is considered as a new design methodology and framework to support distributed problem solving methods in real-time scheduling and optimization of resources [3][4].

The main feature of real-time scheduling and optimization methods is to produce a result in the specified moment of time or time interval, reacting to unpredictable external and internal, constructive or destructive events (new order coming, order is cancelled, resource unavailable, etc.).

The quality and efficiency of decision making in resource scheduling and optimization process can be influenced by the number of factors: the intensity of events flow, the number and current state of resources, individual specifics of orders and resources, time interval between the events and processing time for events, productivity of resources and many others.

A big challenge is to ensure that certain quality of scheduling results is achieved in a short time after the event to make it possible to finish the processing before the next event and to always have a valid schedule needed for decision-making.

Figure 1 illustrates the difference in actuality of scheduling results (how well they reflect reality) in the changing environment. Having frequent data updates, it becomes more important to process them faster to get a valid result (green line). Otherwise, one can use a lengthy processing to get an optimal result (yellow/red line), but this result does not consider the last changes. Then, we are forced to always base your decisions on an optimal, but outdated picture.

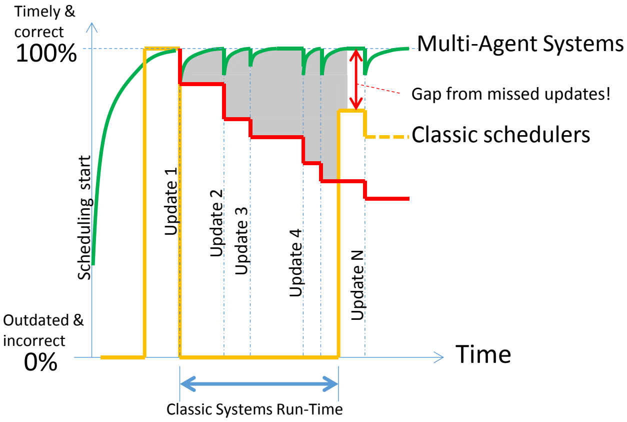


Figure 1. Real-time adaptive scheduling results.

One of the main problems of classical methods and algorithms [5][6] is that complexity of scheduling with new criteria grows exponentially. This makes their applications very limited in practice. Many heuristic methods allow obtaining close to optimal solution within a reasonable time. Hybrid heuristic algorithms integrate traditional dispatching rules with genetic, neural, swarm and other approaches. Obvious disadvantages of the centralized methods of scheduling and optimization resource management lead to development other approaches, in particular distributed problem solving methods. Bio-inspired evolutionary (genetic and swarm) algorithms are applied both in centralized and decentralized systems [7]-[9]. They have proved to be more useful, reliable and generic scheduling and optimization tool for business. One of new approaches is based on bio-inspired distributed problem solving of resource scheduling problems based on multi-agent technologies with economic reasoning. This approach can combine benefits of bio-inspired, DCOP and virtual market methods based on multi-agent technology

ISBN: 978-1-61208-405-3

and is designed to support self-organization of schedules to provide flexibility in event processing. Multi-agent resource allocation is used for job scheduling and some other tasks [11]. In our paper, we consider a more specific practical case of supply scheduling and compare the interaction approaches from the perspective of their use in real-time application. There are other researches done regarding the use of multi-agent approach in supply chain scheduling, including analysis of high-level protocols (Combinatorial Auctions, Bargaining Processes, Random Search, Knowledge Based Systems, Learning Systems) [12][13], but they do not focus on the analysis of benefits of different agent assumptions in asynchronous environment.

To solve the problem of multi criteria scheduling and optimization it is suggested to use Demand-Resource Network concept (DRN) based on holonic approach and compensation method for real-time resource management on a virtual market [10]. In accordance with this distributed approach, initial complex problem is decomposed into more simple and specific problems – to schedule and optimize orders, resources and products with the use of demand and supply agents. All agents are working continuously trying to maximize their objective functions and use money to solve conflicts by negotiations and finding trade-offs (until local optimum is reached or time is expired) with compensations in case that some of them change position losing money.

Objectives, preferences and constraints of agents are defined by individual satisfaction functions and bonus/penalty functions. As the result of agents decision making, a local balance is reached and situation when no agent can change position is recognize as a consensus which stops computations. As a result, the solutions (the schedule of resource usage) comes not from one algorithm but evolves (emerges) dynamically in process of agents interactions and negotiations. Solution search and adjustment process stop when the consensus is found or when the time limit is exceeded for finding a solution, and if not the whole - but partial problem solution will arrive that will be interactively finalized by the user.

The use of multi-agent approach provides many potential benefits and possibility to speed up the scheduling by use of parallel processing of asynchronous agents. Still, this possibility depends on how the agents interact with each other and on their dependence on each other in decision making. Obviously, the scheduling task requires a lot of information to be transferred between the agents to allow a better search for result. This transfer not only takes time itself, but also may force the agents to wait each other. In this paper, we consider two fundamental approaches to agent interactions related to the question when the agent should ask or wait for information, and when it can make independent decisions.

In Section II, we describe what approaches to agent interactions we consider in this paper. In Section III, we compare the interaction schemas based on particular supply routing example. In Section IV, we show how the lack of resources in the supply network affects the interactions, performance and quality of results, and propose the ways to mitigate the drawbacks. In Section V, we compare the approaches based on a more complex case of competing orders in supply network. In Section VI, we provide the results of comparison based on real supply network data, including the difference in performance and quality of the approaches. In Section VII, the conclusion is given.

II. APPROACHES TO AGENT INTERACTION IN SUPPLY SCHEDULING

In this paper, we compare two different approaches to the organization of multi-agent interaction in relation to the supply scheduling. One approach is based on request and reply and follows the rejection presumption principle, which means that if no reply is given it is an equivalent of rejection (sender must wait for an answer). This approach is referred to as rejection assumed interaction in the paper. Another approach is based on the acceptance presumption principle, which means that without explicit rejection from the counterpart of communication the acceptance of request is assumed. This approach is referred to as acceptance assumed interaction in the paper. Of course, this relates to the requests that do not require an informational feedback, but only ask another site to do something, while the feedback is optional.

Let us consider the difference based on a simple example of a network consisting of one shop and two storages that can supply it (Figure 2 and Table I).

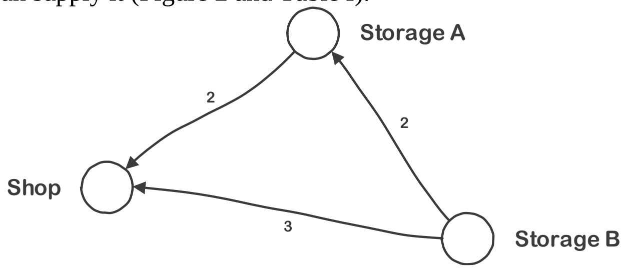


Figure 2. Example of supply network.

There is an order at the Shop for one item of Product. Transportation costs are listed in Table I, and there are no other costs.

TABLE I. TRANSPORTATION COSTS

Source	Destination	Cost per item
Storage A	Shop	2
Storage B	Shop	3
Storage B	Storage A	2

In the simplest example, we have enough stock at both storages. The rejection assumed interaction looks as the following, in this case (Figure 3).

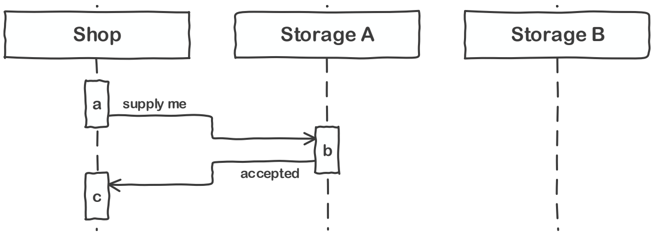


Figure 3. Interaction based on rejection presumption.

It is an obvious case. The order at the shop requests the cheapest channel (channel from Storage A costs 2 while the other channel is 3) if the Product can be delivered and gets the positive reply. The interaction takes three steps in total. Two of them ('a', 'b') are time consuming, as they may require some analysis, while 'c' does nothing, but still takes some time to initialize the agent and process the message. For the sake of simplicity, let us decide that steps with analysis take 1.0 time unit, while steps without significant data processing take 0.1 time units (tu). In this case, the total is 2.1 tu.

If we consider the acceptance assumed interaction for this case, the only difference is that we do not need the last step (Figure 4), as we assume the request is accepted and the supply is possible. Therefore, the total time for processing is 2.0 tu.

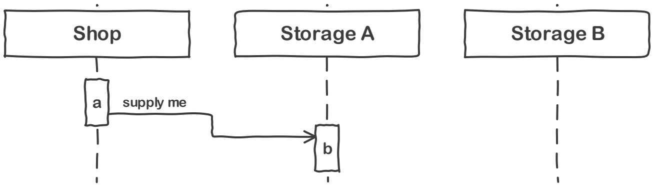


Figure 4. Interaction based on acceptance presumption.

It is important to note that even if an additional processing is needed at the shop to obtain a final result (schedule) after the supply request is considered accepted, in acceptance presumption case this happens immediately after the request is sent and does not take additional time, as being done in parallel with the request processing at the storage.

III. Agent interaction in supply routing scenario

Now, let us consider a less trivial case, where Storage A is empty. The order at the Shop does not have this information and still asks it first in the hope to get cheaper supply. This leads to the following sequence of interactions.

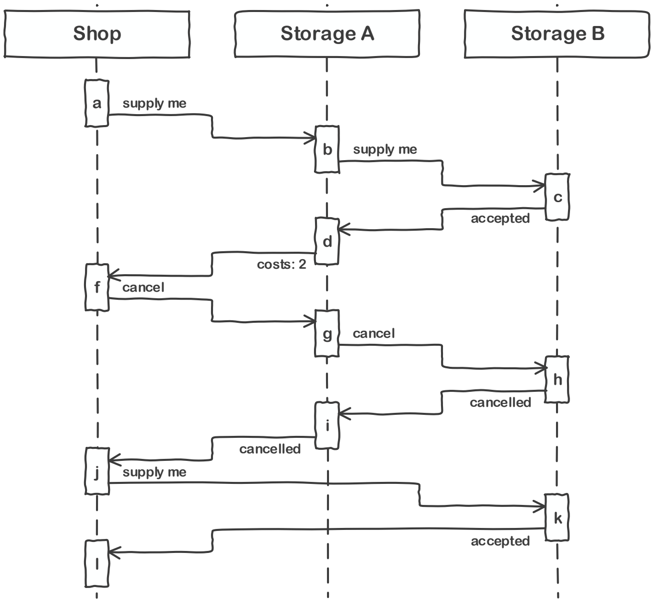


Figure 5. Routing based on rejection presumption.

Storage A, in this case, at step 'b', cannot fulfill the request and sends a supply request to Storage B. At step 'c', Storage B reserves the stock (creates its own schedule) and sends the acceptance. Then, on step 'd' the Storage A sends the acceptance with the additional cost of transportation from B to A. This actually tells the order at the Shop that the total cost of supply will be 2 (from A to Shop) + 2 (from B to A) = 4. This is more than the cost of transportation from B to Shop, which is 3. This makes the order to try another channel. It cancels the previous request (in order to let A and B update their schedule and free the reserved stock) and asks the Storage B directly. The whole interaction takes 11 steps, with four of them ('d', 'i', 'j', 'l') being just fast reply processing. Therefore, the total scheduling time is 7.4 tu. It might be unclear why the steps 'j' and 'l' are "short". It is because we consider the scheduling process at Shop to be almost completely done at step 'f'. When Shop gets the cost reply from Storage A, it has to re-build the schedule to be supplied from Storage B. This may happen in slightly different ways across the steps 'f', 'j', and 'l', but the total re-scheduling time at Shop is assumed to be 1 tu in average, and we just associate this time with the step 'f'.

If we use the acceptance assumed interaction, we get a significantly different picture (Figure 6).

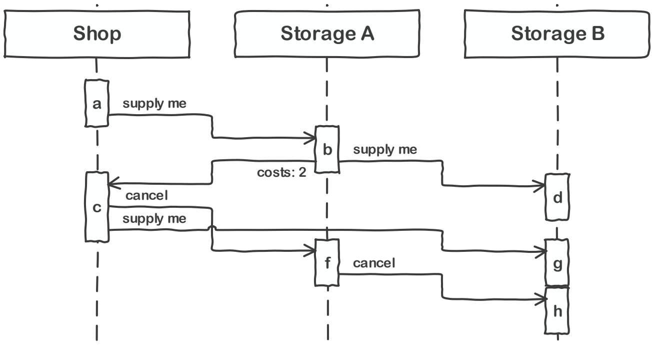


Figure 6. Routing based on acceptance presumption.

We have 7 steps here in total, but all of them are time consuming. More important is that the messages are sent to several recipients at steps 'b' and 'c' as we do not wait for reply, and the corresponding sites process them in parallel. The steps 'c' and 'd' go in parallel, as well as 'f' and 'g'. This allows packing of all the 7 steps into 5.0 tu instead of 7.4 tu of synchronous interaction.

It is important to note that rejection assumed interaction does not mean synchronous processing (scheduling, in our case). There are still things you can do in parallel. For example, interactions happening in different parts of the network can go in parallel. However, with the increasing number of events to be processed also the likelihood of touching the same site increases. If this happens, we need to wait until the first event is processed completely.

IV. Agent interaction in resorce discovery scenario

However, there is a drawback in acceptance assumed interaction, which is clearly seen on the following example. Let us take the case, where the stock at the Storage B is limited to just one item. If we use rejection presumption, the scheduling process and the result are exactly the same as if the stock is not limited, which is correct. But, if the acceptance presumption is used, we get the following interactions (Figure 7).

At step ‘g’, it happens that the one item in the stock of the Storage B is reserved for the Storage A. The reservation takes place at step ‘d’, and the cancellation is now on its way, but the direct request comes faster due to asynchronous nature. This forces the Storage B to reject the direct request from the Shop. Consequently, the Shop asks Storage A again. This not only adds the steps to the interaction and 2.0 tu of processing time, but also leads to the non-optimal result. The Shop is supplied, but the cost is higher, because it is supplied indirectly.

This non-optimality can result in a lower quality of the final schedule, but it happens only in the specific situation when there is lack of stock and re-negotiation between sites takes place at the same time. Even if rejection presumption is used, similar situation still can happen when several orders compete asynchronously. In practical cases, the lack of stock affects a very small part of the orders and the decrease in scheduling is normally acceptable, as the whole scheduling process does not normally achieve a global optimum. Still, this decrease can be a problem in some cases.

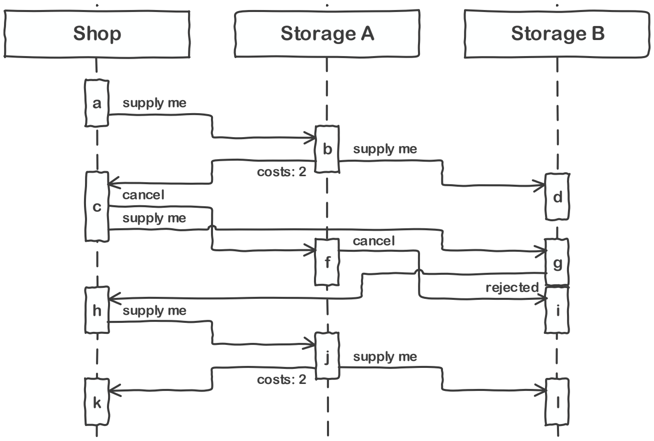


Figure 7. Resource discovery based on acceptance presumption.

Fortunately, there are ways to avoid this problem. One of them is to postpone rejection. With this approach, the site requested for supply does not send the rejection immediately, but waits for the specific number of steps (to be decided later). If the rejection is still needed after several steps (no other requests were cancelled) – it is sent. If we do so in the last case, the rejection is not sent on step ‘g’, and on step ‘i’ it is no longer needed. The scheduling goes in the same way as in the unlimited stock case (no rejection is sent because the cancellation comes at step ‘h’) taking 5.0 tu and optimal result (Figure 6). The weakness of this method is in the necessity to specify the number of steps to postpone the rejection. The number should be high enough to let all the asynchronous cancellations to come before the rejection should be sent. This depends on the structure of the network. And, in the cases when the request really should be rejected the postponement increases the total scheduling time by the number of the specified postponement steps. But the rejection postponement is needed only when the stock is short, which is less than 5% of the requests during the scheduling and does not affect the total timing significantly.

Another method is to track the final product recipient (root order) in the messages so that the site (Storage B, in our case) knows whom the stock is reserved to. If it gets a new request for the same root order, the previous reservation is cancelled automatically. This method also works perfectly and does not depend on the network structure. But the need to check the reservations in the stock for specific order increases the scheduling time at the site. We use this method in practice due to its simplicity and reliability.

V. Agent Interaction in Competing Orders Scenario

The next two examples concern the comparison of the interaction protocols in the situation when there are several competing orders in the network. Taking the case where we have empty stock at Storage A and sufficient stock at Storage B, let us introduce one more sales point in the network to make it look like on the following picture. Now, we have an order at each of the two shops (Figure 8 and Table II).

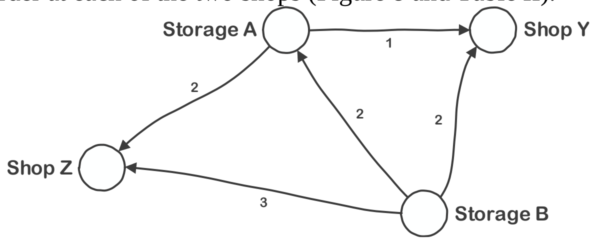


Figure 8. Supply network with two competing orders.

TABLE II. Transportation Costs in the Network with Two Competing Orders

Source	Destination	Cost per item
Storage A	Shop Z	2
Storage B	Shop Z	3
Storage B	Storage A	2
Storage A	Shop Y	1
Storage B	Shop Y	2

Following the rejection presumption principle, the sites cannot process next request until they get a response from other sites regarding the previous request. Thus, in our case the Storage A becomes a bottleneck because both shops ask it first (as potentially cheaper source), and it cannot answer them both until Storage B answers the request. For example, the request processing is blocked at the Storage A on the steps ‘c’, ‘d’, and ‘e’ on the following diagram (Figure 9), which leads to the delay of the processing of the request from the Shop Y on step ‘g’.

Specifically, when Storage A gets a request from Shop Z at step ‘c’, it sends a request for this product to Storage B (as it does not have it in the stock). When the request from Shop Y comes (almost the same time as from Shop Z), it cannot be processed until the request to Storage B is accepted.

We consider only one product in the network in the paper, so the orders compete for the same stock. If you get requests for different products, they theoretically may be processed immediately one after another, but this is an abstract situation. In practical tasks there are much more interdependencies between resources and demands other than just product type. For example, the channel capacity between Storage A and Storage B, or the dispatch capacity at Storage B can be limited, or the transportation cost may depend nonlinearly on the volume transported. This prevents Storage A from answering the second request even if it is for a different product, until the acceptance of the first delivery is received (or assumed).

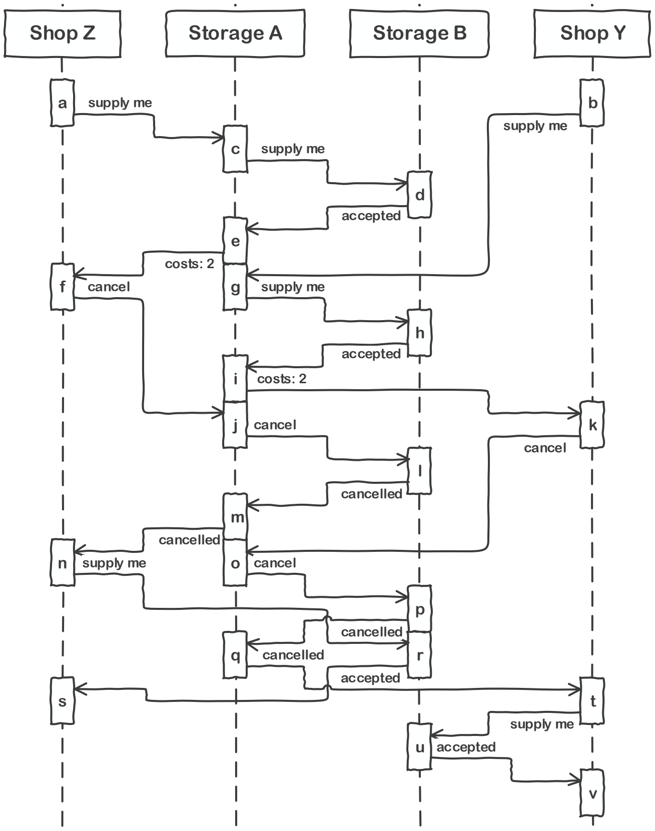


Figure 9. Competitive interactions with rejection presumption.

The complete processing of the two orders with this approach takes 22 steps. Considering that some of them are done in parallel and some of them are very quick, this exact sequence takes 12.4 tu.

Actually, we do not consider here the fully synchronous interaction that requires all events to be processed separately. It means that the order from the Shop Z is completely processed first, and only then the processing of the order from the Shop Y starts. This forces the whole sequence to go in one thread and take 16.6 tu.

The next diagram (Figure 10) shows the interactions using acceptance presumption protocols.

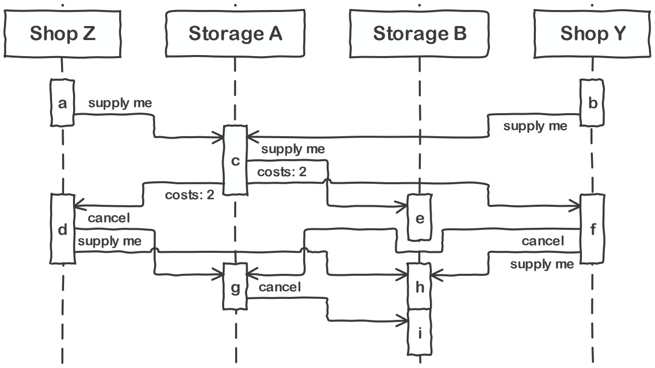


Figure 10. Competitive interactions with acceptance presumption.

One can see that, in this example, the structure of the interactions is the same as in the case where we had only one order. The whole process goes as much as possible in parallel and takes the same 5.0 tu. The significant difference from the rejection presumption case is that at some steps several requests are processed by the site simultaneously. From one point of view, such steps should take more time, but from the other point of view, the processing of several requests at once never takes more time than separate processing of the same requests. What is more important, having several requests at once allows avoiding blind decisions that should be re-considered when the next request comes. A separate paper is dedicated to this phenomenon.

VI. COMPARISON BASED ON REAL DATA

Thus, based on the examples above, the theoretical comparison of the two approaches is shown in Table III.

TABLE III. EXAMPLES SUMMARY

Case	Fully synchronous processing	Rejection presumption (tu)	Acceptance presumption (tu)
One-level depth interaction.	2.1	2.1	2.0
Two-level depth interaction without resource constraints.	7.4	7.4	5.0
Two-level depth interaction with resource constraints.	7.4	7.4	7.0
Two orders, two-level depth interaction.	16.6	12.4	5.0

The practical cases are much more complex in terms of the depth of interactions as well as of the number of events processed in parallel. We used a real client data including more than 300 sites in the network (part of which is fully interconnected) and about 10 000 orders to model different interaction protocols. The network to be scheduled includes several factories, their storages that can interexchange materials and final products, and customer distribution centers that should be supplied. The model also includes production scheduling and some other features that affect the processing time in different situations. The modelling has been done using 16-core processor. Table IV presents the results of the modelling.

TABLE IV. REAL DATA PROCESSING

	Processing time (ms)	**Messages between sites**	**Achieved quality ($)**
Fully synchronous processing	737236	3200	1813499
Rejection presumption	191334	3140	1813359
Acceptance presumption	50275	2333	1812240

The slight difference in quality between the synchronous processing and the rejection presumption most probably happens because of asynchronous stock competition between different orders.

Comparing the last two rows we can see that the use of acceptance presumption approach gives us 3.8 times faster processing and decreases the quality by about 0.1%, which seems to be a fair price in most cases.

VII. CONCLUSION

The acceptance assumed interaction works much better than the rejection presumption in multicore and especially in distributed environments because waiting for reply there is especially costly. However, it is fragile in non-reliable communication environments. If the requested site in the network does not implement the request and does not send the rejection, the requesting site works in wrong assumptions and the whole schedule is not consistent. This is why it can only be used within well-communicated infrastructure, normally related to one company.

We use the acceptance presumption approach in the industrial applications for supply networks management.

It is also important to make a research how the two approaches can be combined in some way during the interaction. Although the acceptance presumption looks working better in most cases, especially below the resource limits, which is over 90% of the practical cases, the rejection presumption may still allow getting results of higher quality without using workarounds in the low resource situations.

ACKNOWLEDGMENT

This work was carried out with the financial support of the Ministry of Education and Science of the Russian Federation.

REFERENCES

[1] G. Rzevski and P. Skobelev, Managing complexity, 2014, WIT Press, Boston.

[2] A. Park, G. Nayyar, and P. Low, Supply Chain Perspectives and Issues, A Literature Review, April 21, 2014, Fung Global Institute and World Trade Organization.

[3] A. Mohammadi and S. Akl, Scheduling Algorithms for Real-Time Systems, Technical Report, 2005, no. 2005-499, School of Computing Queen’s University, Kingston.

[4] M. Joseph, Real-time Systems: Specification, Verification and Analysis, Prentice Hall, 2001.

[5] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, Springer, 2008.

[6] J. Leung, Handbook of Scheduling: Algorithms, Models and Performance Analysis, CRC Computer and Information Science Series, Chapman & Hall, 2004.

[7] S. Binitha and S. Sathya, “A survey of bio inspired optimization algorithms”, Int. Journal of Soft Computing and Engineering, 2012, vol. 2, issue 2, pp. 2231-2307.

[8] S. Sun and J. Li, “A two-swarm cooperative particle swarms optimization”, Swarm and Evolutionary Computation, 2014, vol. 15, pp. 1-18. Elseiver.

[9] M. Tasgetiren, M. Sevkli, Y. Liang, and M. Yenisey, “Particle swarm optimization and differential evolution algorithms for job shop scheduling problem”, International Journal of Operational Research, 2008, vol. 3, no. 2, pp. 120-135.

[10] V. Vittikh and P. Skobelev, “Multiagent interaction models for constructing the demand-resource networks in open systems”, Automation and Remote Control, 2003, vol. 64, issue 1, pp. 162-169.

[11] Y. Chevaleyre, et al. “Issues in Multiagent Resource Allocation”, https://staff.science.uva.nl/u.endriss/MARA/mara-survey.pdf, retrieved: March 2015.

[12] M. Barbuceanu and M. S. Fox, “Coordinating multiple agents in the supply chain”, Proceedings of the fifth workshops on enabling technology for collaborative enterprises, WET ICE'96, IEEE Computer Society Press, 1996, pp. 134-141.

[13] T. Stockheim, M. Schwind, O. Wendt, and S. Grolik. “Coordination of supply webs based on dispositive protocols”, 10th European Conference on Information Systems (ECIS), Gdañsk, 6-8 June 2002.

[14] A. Oliinyk, “The multiagent optimization method with adaptive parameters”, Artificial Intelligence journal, 2011, no. 1, pp. 83-90.

On The Topological Entropy of Continuous-Time Polytopic Systems

Graziano Chesi

Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Email: chesi@eee.hku.hk

***Abstract*—This paper investigates the topological entropy of continuous-time polytopic systems. The topological entropy is a measure that quantifies the instability in dynamical linear systems and has important applications in autonomous systems. Polytopic systems are dynamical linear systems whose coefficients are functions of an uncertain vector constrained into a polytope. A novel approach is proposed for establishing upper bounds of the largest topological entropy of continuous-time polytopic systems based on the Routh-Hurwitz stability criterion. The upper bounds are established through Linear Matrix Inequality (LMI) feasibility tests, which amount to solving convex optimization problems. A numerical example illustrates the proposed approach.**

Keywords–Topological entropy; Polytopic systems; LMI.

I. Introduction

The topological entropy is a measure that quantifies the instability in dynamical linear systems. This measure is defined as the sum of the real part of the unstable eigenvalues in the continuous-time case, and as the product of the magnitude of the unstable eigenvalues in the discrete-time case [1]. The topological entropy has important applications in autonomous systems where it is required to ensure stability with communication constraints [2]. For instance, this measure can be used to establish the existence of stabilizing state feedback controllers in the presence of constraints on the signal-to-noise ratio [3]. See also [4] [5] for other uses of this measure.

Unfortunately, the mathematical model of a control system is often affected by uncertainty, e.g., representing physical quantities that cannot be measured exactly or that are subject to changes. As a consequence, one has to consider a family of admissible models depending on the uncertainty. Clearly, the instability measures become functions of the uncertainty, and the target is to determine the largest instability measures over the admissible uncertainties.

In the literature, the topological entropy of continuous-time uncertain systems has been investigated in [6] [7] through convex optimization. However, these methods exploit Lyapunov functions [8] and determinants, and cannot be easily used for control design because the presence of an unknown controller would lead to the formulation of nonconvex optimization problems.

In order to deal with this drawback, a novel approach is proposed in this paper for investigating the topological entropy of continuous-time uncertain systems. Specifically, polytopic systems are considered, i.e., dynamical linear systems whose coefficients are functions of an uncertain vector constrained into a polytope. It is shown that upper bounds of the largest topological entropy can be established based on the Routh-Hurwitz stability criterion through LMI feasibility tests, which amount to solving convex optimization problems. A numerical example illustrates the proposed approach.

The paper is organized as follows. Section II introduces the preliminaries. Section III describes the proposed results. Section IV presents an illustrative examples. Lastly, Section V concludes the paper with some final remarks.

II. Preliminaries

Notation: $\mathbb{R}, \mathbb{C}$: sets of real and complex numbers; $\Re(M), \Im(M)$: real and imaginary parts of M; I: identity matrix (of size specified by the context); M': transpose; $M > 0$, $M \geq 0$: symmetric positive definite and symmetric positive semidefinite matrix; $\lambda_i(M)$: i-th eigenvalue of M; $\operatorname{spec}(M)$: set of eigenvalues of M; $\|M\|_2$: 2-norm of v; M^2: entry-wise square; Hurwitz matrix: a matrix whose eigenvalues have negative real parts.

Let us consider the continuous-time uncertain system

$$\dot{x}(t) = A(p)x(t) \tag{1}$$

where $t \in \mathbb{R}$ is the time, $x(t) \in \mathbb{R}^n$ is the state, $p \in \mathbb{R}^q$ is an uncertain vector constrained by

$$p \in \mathcal{S} \tag{2}$$

where $\mathcal{S}$ is the simplex

$$\mathcal{S} = \left\{ p \in \mathbb{R}^q : \ p_i \geq 0, \ \sum_{i=1}^{q} p_i = 1 \right\}, \tag{3}$$

and $A(p) \in \mathbb{R}^{n \times n}$ is a matrix polynomial.

Let $B \in \mathbb{R}^{n \times n}$. The topological entropy of B is defined as

$$\mu(B) = \sum_{i=1}^{n} \max\left\{0, \Re(\lambda_i(B))\right\}, \tag{4}$$

i.e., as the sum of the real part of the unstable eigenvalues of B.

Problem 1. The problem that we consider in this paper consists of determining the largest topological entropy of (1)–(3), i.e.,

$$\mu^* = \sup_{p \in \mathcal{S}} \mu(A(p)). \tag{5}$$

ISBN: 978-1-61208-405-3

III. Proposed Approach

The first step of the proposed approach is to introduce a matrix whose eigenvalues are all the possible sums of the eigenvalues of a given matrix. Specifically, let $B \in \mathbb{R}^{n\times n}$ the given matrix, and let $k = 1, \dots, n$ denote the number of eigenvalues that have to be multiplied. We denote with $\Omega_k(B)$ the matrix function whose eigenvalues are all the possible sums of k eigenvalues of B, i.e.,

$$\mathrm{spec}(\Omega_k(B)) \left\{ \sum_{i=1}^{k} \lambda_{z_i}(U),\ z \in \mathcal{I}_k \right\} \tag{6}$$

where $\mathcal{I}_k$ is the set of k-tuples in $\{1, \dots, n\}$ defined by

$$\begin{aligned} \mathcal{I}_k \quad = \quad & \{(z_1, \dots, z_k) :\ z_i \in \{1, \dots, n\}, \\ & z_i < z_{i+1}\ \forall i = 1, \dots, k-1\}. \end{aligned} \tag{7}$$

The matrix function $\Omega_k(B)$ can be built for any positive integer n and for any $k \in \{1, \dots, n\}$ following the idea described by Bellman [9].

The second step of the proposed approach is to build a modified Routh-Hurwitz table. Specifically, let $B \in \mathbb{R}^{n\times n}$ and $\theta \in \mathbb{C}$. Let us define

$$f(\theta, B) = \det(\theta I - B) \tag{8}$$

which is a polynomial in θ. We denote with $g_{i,j}(B)$ the (i,j)-th entry of the table obtained for $f(\theta, B)$ under the following constraints:

1) $g_{i,j}(B)$ is a polynomial in B;
2) $g_{i,1}(B) > 0$ if and only if B is Hurwitz.

The third step of the proposed approach is to exploit convex optimization. Specifically, let $w > 0$, and for $k = 1, \dots, n$ let us define

$$h_{i,k}(p, w) = g_{2i,1}\left(\Omega_k(A(p)) - wI\right) \tag{9}$$

which is a polynomial in p. Let $m_{i,k}(p, w)$ be the homogeneous polynomial in p with the minimum degree satisfying

$$m_{i,k}(p, w) = h_{i,k}(p, w)\ \ \forall p \in \mathcal{S}. \tag{10}$$

Let $d_{i,k}$ denote such a degree. Then, a condition for establishing that w is an upper bound of μ^* can be obtained by looking for a scalar $\varepsilon > 0$ such that

$$m_{i,k}(p^2, w) - \varepsilon \|p\|_2^{2d_{i,k}} \text{ is SOS } \forall i, k \tag{11}$$

where SOS stands for sum of squares of polynomials.

The condition (11) amounts to solving a convex optimization problem because establishing whether a polynomial is SOS amounts to establishing feasibility of an LMI; see, for instance, [10] and references therein. It can be shown that the condition (11) is sufficient for establishing that w is an upper bound of μ^*. Moreover, it can also be shown that this condition is also necessary by suitably increasing the degree of the polynomial in (11) following the ideas in [11].

IV. Example

Let us consider for simplicity (1)–(3) with

$$A(p) = p_1 \begin{pmatrix} 3.4 & 2.9 \\ -1.6 & -1.6 \end{pmatrix} + p_2 \begin{pmatrix} -2.9 & -4.1 \\ 4.5 & 0.3 \end{pmatrix}.$$

We test the condition (11) for different values through bisection, finding that the best upper bound guaranteed by this condition is

$$\hat{\mu} = 2.457$$

(the condition (11) is equivalent to an LMI with 4 scalar variables and can be solved in less than one second on standard personal computers). Brute force search shows that this upper bound is tight, i.e., $\mu^* = \hat{\mu}$. Indeed,

$$p = (0.785, 0.215)' \quad \Rightarrow \quad A(p) = \hat{\mu}.$$

V. Conclusion

A novel approach has been proposed for establishing upper bounds of the largest topological entropy in continuous-time polytopic systems. The proposed approach can be easily implemented with standard software, moreover the numerical example has shown that the computational burden can be significantly low and that the upper bounds can be non-conservative. Future work will explore the use of the proposed approach for control design.

Acknowledgement

This work is supported in part by the Research Grants Council of Hong Kong under Grant HKU711213E.

References

[1] R. Bowen, “Entropy for group endomorphisms and homogeneous space,” Transactions of the American Mathematical Society, vol. 153, 1971, pp. 401–414.

[2] G. N. Nair, F. Fagnani, S. Sampieri, and R. J. Evans, “Feedback control under data rate constraints: an overview,” Proceeding of IEEE vol. 95, 2007, pp. 108–137.

[3] J. H. Braslavsky, R. H. Middleton, and J. S. Freudenberg, “Feedback stabilization over signal-to-noise ratio constrained channels,” IEEE Transactions on Automatic Control, vol. 52, no. 8, 2007, pp. 1391–1403.

[4] M. Fu and L. Xie, “The sector bound approach to quantized feedback control,” IEEE Transactions on Automatic Control, vol. 50, no. 11, 2005, pp. 1698–1711.

[5] L. Qiu, G. Gu, and W. Chen, “Stabilization of networked multi-input systems with channel resource allocation,” IEEE Transactions on Automatic Control, vol. 58, no. 3, 2013, pp. 554–568.

[6] G. Chesi, “Measuring the instability in continuous-time linear systems with polytopic uncertainty,” in IEEE Conference on Decision and Control, Florence, Italy, 2013, pp. 1131–1136.

[7] G. Chesi, “LMI-based computation of the instability measure of continuous-time linear systems with a scalar parameter,” in IEEE Canadian Conference on Electrical and Computer Engineering, Toronto, Canada, 2014, pp. 374–379.

[8] J. P. LaSalle and S. Lefschetz, Stability by Lyapunov’s Direct Method with Applications. New York: Academic Press, 1961.

[9] R. Bellman, Introduction to Matrix Analysis. New York: McGraw-Hill, 1974.

[10] G. Chesi, “LMI techniques for optimization over polynomials in control: a survey,” IEEE Transactions on Automatic Control, vol. 55, no. 11, 2010, pp. 2500–2510.

[11] G. Chesi, “On the non-conservatism of a novel LMI relaxation for robust analysis of polytopic systems,” Automatica, vol. 44, no. 11, 2008, pp. 2973–2976.

Task-Space Torque Controller Based on Time-Delay Control

Sung-moon Hur*, Joonhee Jo*†, Yonghwan Oh*
*Center for Robotics Research, Korea Institute of Science and Technology
Hwarang-ro 14-gil 5 Seongbukgu Seoul, Korea
Email: tjdans85@gmail.com, oyh@kist.re.kr
†HCI and Robotics, University of Science and Technology
KIST Campus, Seoul, Korea
Email: jhjo@kist.re.kr

***Abstract*—In this paper, a torque control method in task-space for redundant manipulators with friction is proposed. A previous simple control approach based on virtual spring damper hypothesis is used to generate human-like motions. The method is efficient in the system which is difficult to identify the exact dynamics, however, the controller has steady state errors. To eliminate the steady state error, the gravity and friction, which is the part of the system dynamics, are compensated. Although the gravity and friction are compensated, the error of the modelling remains in the system. Hence, to reduce the nonlinearity, unknown effects, and modelling errors of the system, a torque controller based on Time-Delay Control (TDC) that eliminates the friction and unknown effects, is used. The performance of the control method, in Cartesian space control, is experimented with the torque sensor based 3-joints robot manipulator.**

Keywords–Task-Space; Virtual spring damper hypothesis; Time-Delay Control(TDC).

I. INTRODUCTION

Recently, many robot research for redundant manipulators have been developed, and to control the robot precisely in task-space has been an issue, especially for industrial robots. The traditional control method is to compute the inverse kinematics of the system [1][2][3]. The control input is computed from the joint angle velocity, which is calculated from the given end-effector velocity. Another approach is to create the control input directly from the inertia matrix and Coriolis and centrifugal force, which is called the inverse dynamics approach [4]. However, these control methods are especially difficult to compute in redundant systems because of the calculation for pseudo-inverse of the Jacobian matrix. Therefore, a simple approach that does not need for any computation of the inverse kinematics nor dynamics which is proposed by Arimoto *et al*. is considered. This is a natural control method based on virtual spring-damper hypothesis [7]-[11], which offers human-like motions. In this paper, the natural control method based on virtual spring damper hypothesis is used for the task-space controller.

For precise control of the end-effector, the dynamic model of the system is required. However, non-linearity of the system makes it difficult to model and causes control problems. The non-linearity of the system is the friction from the harmonic drive and bearing, noise and flexibility of the sensor, and dynamic modelling error of the plant. To deal with these problems many researches have been proposed such as using an observer to estimate the disturbance [13][24][25], friction compensation method [12], Time-Delay Control (TDC) method that eliminates the uncertainties without using the system parameters [14][15][16][18][19], and impedance control [6].

This paper addresses a task-space torque control method to control the friction existing redundant manipulator accurately. To overcome the friction and uncertainties, the TDC method is applied. With the TDC method used, the torque controller estimates the non-linear friction, unknown effects, and dynamic errors and cancel them out without any parameter identification [14].

This paper is organized as follows. The dynamic model of the redundant manipulator and the friction model is handled in Section II, and the task-space torque controller is designed in Section III. Experiments are carried on to validate the proposed method in Section IV, and Section V concludes this paper.

II. DYNAMIC MODEL OF THE REDUNDANT MANIPULATOR

A. Dynamic Model

The dynamic model of the redundant manipulator is considered as [20]

$$M(q)\ddot{q}+c(q,\dot{q})q+g(q)=\tau_o+\tau_{ext} \tag{1}$$

$$B\ddot{\theta}+\tau_o=\tau_m+\tau_f \tag{2}$$

$$\tau_o=k_s(\frac{\theta}{N}-q)+k_d(\frac{\dot{\theta}}{N}-\dot{q}) \tag{3}$$

where (1) is the model of the link, (2) is the model of the motor, and (3) is the joint torque of the manipulator. q is the link angle vector, θ is motor angle vector, $M(q)$ is the mass inertia matrix of the manipulator, $c(q,\dot{q})$ Coriolis force, $g(q)$ is the gravity, τ_{ext} is the vector of the friction and external disturbance, τ_o is the torque measured by the joint torque sensor, B is the motor inertia, τ_f stands for the friction torque of the motor, k_s, k_d, N, and τ_m depict is the joint stiffness, joint damping, gear ratio, and the motor input torque,respectively. Each joint torque is measured by the joint torque sensor installed in each joint of the manipulator.

B. Friction Model

Friction, from the harmonic drives and bearings, causes control problems [18][21][22][23]. In velocity or position servo, the friction can be ignored by the appropriately chosen gain of the controller. However, with joint torque servo, friction lowers the performance of the system, as shown by Hur *et al*. [18]. To improve the control performance and make a margin of gain the friction should be compensated with a appropriate model. The friction is identified with a simple experiment with the concept that the friction depends on the velocity and torque of the joints [19]. Based on the Coulomb Viscous friction model [22] the 3 joint manipulator friction model is estimated as

$$\tau_f = \hat{\tau}_c \tanh(\alpha\dot{\theta}) + \hat{\tau}_v \tanh(\alpha\dot{\theta})\sqrt{\left|\dot{\theta}\right|} \tag{4}$$

where α is the slope of the ramp, $\hat{\tau}_c$ denotes the Coulomb friction parameter, and $\hat{\tau}_v$ represents the Viscous friction parameter. With the estimated friction model of the 3-joints robot manipulator the friction is compensated by adding to the control input. From the experiment, the friction is estimated; however, because of the Coulomb friction compensator the system starts to chatter [18]. To eliminate the static friction while ensuring the stability of the system a stiction feed-forward method is applied. The static friction is expressed as

$$\tau_{st} = \begin{cases} +\tau_s & for\ |\dot{q}| \ \geq \ \dot{q}_b,\ \beta_e > \beta_{eb} \\ -\tau_s & for\ |\dot{q}| \ \geq \ \dot{q}_b,\ \beta_e < -\beta_{eb} \\ 0 & others \end{cases} \tag{5}$$

where τ_{st} is the static feed-forward parameter, τ_s stand for the static friction coefficient, $\dot{q}_b$ is the velocity boundary, β_e is the error of the control target, β_{eb} symbolizes the control error boundary. The boundary of the joint velocity and the control error is decided from amplitude of the sensor noise at steady state. The final friction model of the system is as follow.

$$\tau_f = \hat{\tau}_c \tanh(\alpha\dot{\theta}) + \hat{\tau}_v \tanh(\alpha\dot{\theta})\sqrt{\left|\dot{\theta}\right|} + \tau_{st}. \tag{6}$$

III. TASK-SPACE CONTROLLER

To control the redundant manipulator a traditional method is an inverse dynamics approach when the dynamic equation is as (1).

$$\tau = M(q)J^{+}(q)(\ddot{X}_c - \dot{J}(q,\dot{q})\dot{q}) + C(q,\dot{q}) + g(q) \tag{7}$$

where J^{+} is the pseudo inverse matrix, and X denotes the Cartesian position vector of the task-space

$$X = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}, \tag{8}$$

and $\ddot{X}_c$ is the task-space command acceleration and is expressed as

$$\ddot{X}_c = \ddot{X}_d + K_v(\dot{X}_d - \dot{X}) + K_p(X_d - X) \tag{9}$$

where X_d, $\dot{X}_d$, $\ddot{X}_d$ are the desired Cartesian position, velocity, and acceleration, and K_v, K_p are the gain matrices. In an ideal condition, the controller follows the error dynamics

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{10}$$

where $e = X_d - X$, the Cartesian position error; however, to use this method the inverse jacobian matrix needed and makes the system complicated. To simplify the controller a natural control method based on virtual spring damper hypothesis is used.

A. Virtual Spring Damper Hypothesis

In this paper, the manipulation controller for the redundant manipulator to obtain a human-like motion is based on the virtual spring damper hypothesis which is suggested in [5]. This is a simple control approach which does not need calculation of the pseudo-inverse of the jacobian matrix or the dynamics of the system. With the Cartesian position error e the virtual spring potential energy U is as [17]

$$U = \frac{1}{2}e^T K e \tag{11}$$

where K is the stiffness coefficients matrix of the end-effector. The potential energy U is derivative in time $\frac{dU}{dt}$ to obtain the torque and joint velocity. When $\dot{X} = J(q)\dot{q}$

$$\frac{dU}{dt} = e^T K\dot{e} = -e^T K J(q)\dot{q} = -\tau^T\dot{q} \tag{12}$$

From (12), the torque is

$$\tau = -J^T(q)Ke. \tag{13}$$

With the torque (13), the virtual spring hypothesis is expressed as

$$\tau = -C\dot{q} - J^T(q)Ke \tag{14}$$

where C represent the damping coefficients matrix of the joint. For human like motions, the joint damper,$-C\dot{q}$, occur over-damping problem and to show similar movements a virtual damper, $K_v\dot{X}$ is added with the virtual spring and is extended to virtual spring damper hypothesis.

$$\tau = -C\dot{q} - J^T(q)(K_v\dot{X} + Ke) \tag{15}$$

where K_v denotes the damping coefficient matrix. To improve the controller performance, a friction and gravity compensator is considered with the virtual spring damper controller and is as

$$\tau = -C\dot{q} - J^T(q)(K_v\dot{X} + Ke) + \tau_f + g(q). \tag{16}$$

B. Torque Controller Based on Time Delay Control

As shown in [19], the friction can be estimated by using the concept that the friction is related with velocity and torque, yet it is difficult to identify the non-linear phenomena and unknown effects. To eliminate the friction, non-linear and unknown effects without additional experiments, a torque controller based on TDC method is proposed [15]. The non-linear and linear time invariant system is defined to consider the TDC method.

$$\dot{x} = f(x,t) + B(x,t)u + d(t) \tag{17}$$

$$\dot{x}_m = A_m x_m + B_m r \tag{18}$$

where x denotes the state vector, t is the time, $f(x,t)$ the full dynamics of the robot which includes the non-linear and unknown effects, $B(x,t)$ control distribution, u is the control input, and $d(t)$ is the external disturbance. x_m represents the state vector of the reference model, A_m system matrix, B_m is the command distribution matrix, and r is the command vector. The linear time invariant system (18) is a system without friction or disturbance. The non-linear systems control input u is defined by the error dynamics, $\dot{e} = A_m e$, to be controlled as the linear time invariant system where $e = x_m - x$. By (17) and (18) the error dynamics is as

$$\dot{e} = A_m e + \{-f(x,t) - B(x,t)u - d(t) + A_m x_m + B_m r\}. \tag{19}$$

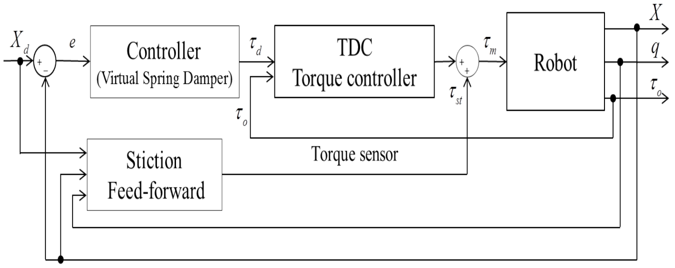


Figure 1. Block diagram of Task-Space Controller with TDC based Torque Controller

When the non-linear term is as

$$-f(x,t) - B(x,t)u - d(t) + A_m x_m + B_m r = 0$$

the system is stable; hence, the control input is derived as

$$u = B^+[-f(x,t) - d(t) + A_m x_m + B_m r] \tag{20}$$

where B^+ is the pseudo-inverse of B, which is $B^+ = (B^T B)^{-1} B^T$.

The assumption of the TDC method is that the present time value is the same after a very short time δ have passed. It is expressed as

$$f(x,t) + d(t) \cong f(x,t-\delta) + d(t-\delta) \tag{21}$$

From the assumption the non-linear effects are estimated.

$$f(x,t) + d(t) \cong \dot{x}(t-\delta) - B(x,t-\delta)u(t-\delta) + d(t-\delta). \tag{22}$$

Substituting (22) into the control input u the TDC control input is as follow:

$$u(t) = u(t-\delta) + B^+[-\dot{x}(t-\delta) + A_m x + B_m r]. \tag{23}$$

For the torque controller of this paper based on TDC control law, control input, u, is defined as

$$\tau_m(t) = \tau_m(t-\delta) + \hat{M}[-\ddot{\tau}_d(t-\delta) + \ddot{\tau}_d(t) + k_p\tau_e + k_v\dot{\tau}_e]. \tag{24}$$

where τ_d stand for the desired torque, $\hat{M}$ denotes a constant diagonal matrix followed by the stability analysis [16], τ_e is the torque error $\tau_e = \tau_d - \tau_o$. Although the TDC is a controller that eliminates the non-linearity effects there is a limitation on canceling the static friction. Therefore, the static friction compensator is applied to improve the control performance. The overall task-space controller is described in Figure 1.

IV. Experiments

The redundant manipulator that is used in this paper is a 3-joints robot arm equipped with a joint torque sensor at each joint. Each parameters of the link mass, length, the position of the center of mass, mass moments of inertia are shown in Table 1. With the proposed approach, the performance controlling the manipulator in task-space by eliminating the friction and unknown effects is tested in this section. The actuators are controlled by the motor controller from ELMO, and the harmonic drives are directly connected to the actuators with a 101:1 gear ratio.

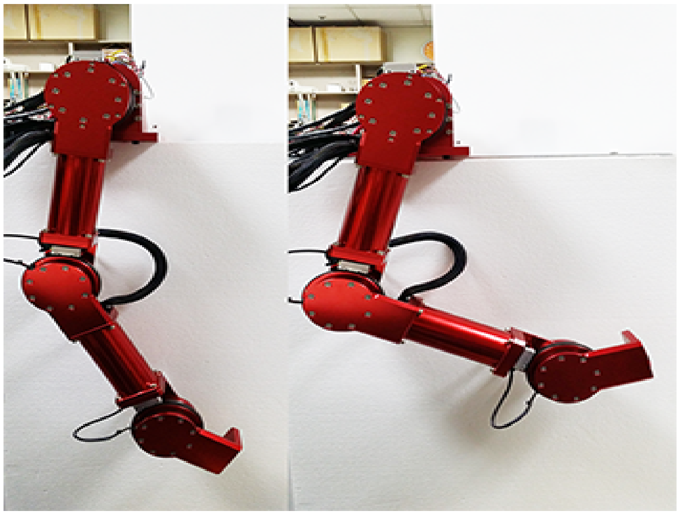


Figure 2. Initial Position, $X = [0.150, -0.550]^T$, and Target Point, $X = [0.300, -0.400]^T$, of the 3-Joint robot arm

TABLE I. 3-Joint Robot Arm Parameters

Parameters	*Value*	*Unit*
m_1	2.3292	*kg*
m_2	2.2589	*kg*
m_3	2.0013	*kg*
l_1	0.300	*m*
l_2	0.300	*m*
l_3	0.146	*m*
l_{c1}	0.13552	*m*
l_{c2}	0.14023	*m*
l_{c3}	0.06857	*m*
I_1	0.041629	kgm^2
I_2	0.039832	kgm^2
I_3	0.0082305	kgm^2

A. Task-Space Controller without Friction Compensator

The task-space controller, which is based on virtual spring damper, is experimented. In this experiment, the control input is as

$$\tau = -C\dot{q} - J^T(q)(K_v\dot{X} + Ke) + g(q). \tag{25}$$

To verify the controller performance the experiment starts at $X = [0.150, -0.550]^T$, point 1, and moves to $X = [0.300, -0.400]^T$, point 2, and comes back to point 1 as Figure 2. Figure 3 is the experiment result of the controller with the 3-joint manipulator. From the end-effector position of Figure 3, the red line is the desired position of the $x-axis$, and the blue line is the $x-axis$ position of the end-effector. The magenta line represents the desired position of the $y-axis$, and the green line is the $y-axis$ position of the end-effector. In the end-effector error graph, the red line, x_{err}, is the $x-axis$ position error, and the blue line, y_{err} is the $y-axis$ position error. The errors are calculated between the desired and current position of each axis. From the result of the task-space controller without friction compensation, it converges to the desired position with approximately $x-axis$ $0.01[m]$, and $y-axis$ $0.01[m]$ error. The error comes from the modelling error, non-liner friction, and the unknown effects, therefore, the next experiment include the friction of each joint and is compensated. Next, the task-space controller with the friction compensator is tested.

ISBN: 978-1-61208-405-3

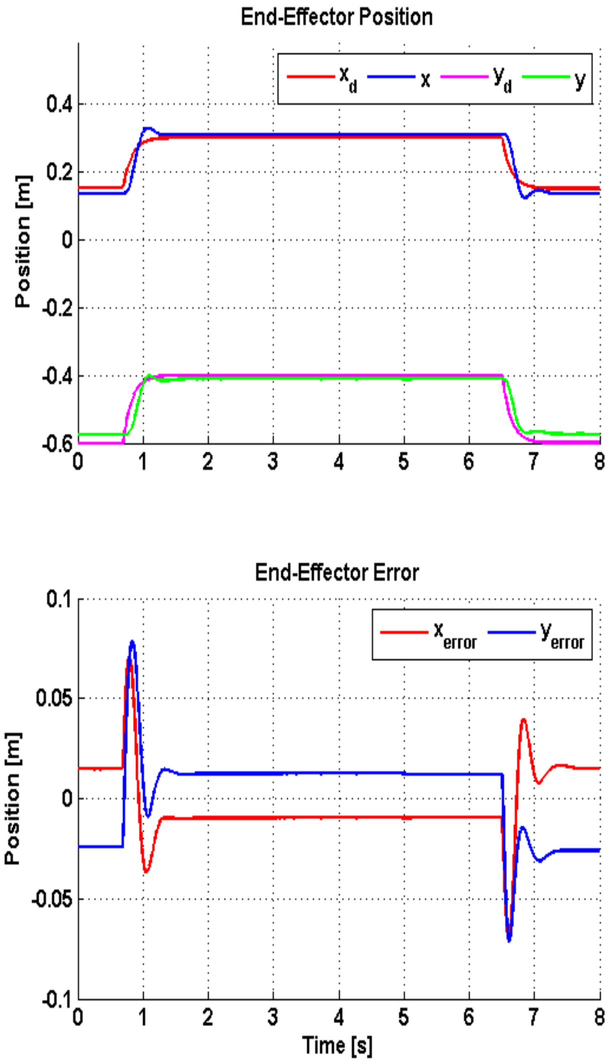


Figure 3. Control Result with Virtual Spring Damper Hypothesis

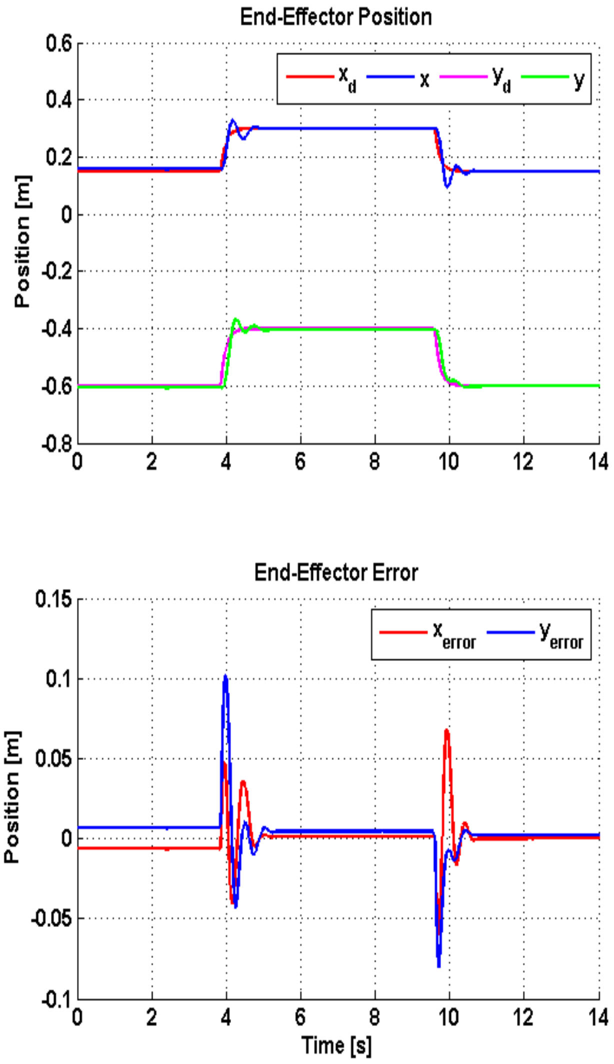


Figure 4. Control Result with Virtual Spring Damper Hypothesis and Friction Compensation

B. Task-Space Controller with Friction Compensator

Without considering friction as the early experiment, from the experimental result friction degrades the performance of the system. Therefore, by implementing the friction compensator to the task-space controller, better performance of the controller is achieved. The control input of this experiment is as

$$\tau = -C\dot{q} - J^T(q)(K_v\dot{X} + Ke) + \tau_f + g(q). \qquad (26)$$

Figure 4 shows the result of the controller with the friction compensator and the lines denotes the same as Figure 3. By applying the friction model, the position error decreases to approximately $x-axis$ $0.001[m]$, and $y-axis$ $0.004[m]$. With the friction compensator an appropriate performance of the controller is obtained. From the result of Figure 4, it appears that the controller performance depends on how the friction and the system is modeled; however, these models are difficult to identify. A TDC control method, as explained at the early section, is used to treat the difficulties in the next experiment.

C. Task-Space Torque Controller Based on TDC

To eliminate the non-linear friction, unknown effects, and modelling error a TDC torque controller is used instead using the estimated friction model. This method needs no additional experiments for identifying the friction nor system dynamic parameters and is very adaptive in friction existing systems. The torque control input in task-space control is designed as

$$\begin{aligned} \tau_d &= -J^T(q)(K_v\dot{X} + Ke) \\ \tau_m(t) &= \tau_m(t-\delta) \qquad (27) \\ &\quad + \hat{M}[-\ddot{\tau}_d(t-\delta) + \ddot{\tau}_d(t) + k_p\tau_e + k_v\dot{\tau}_e] - C\dot{q}. \end{aligned}$$

From the result in Figure 5, the lines have the same meaning as in Figure 2. The result shows that the controller

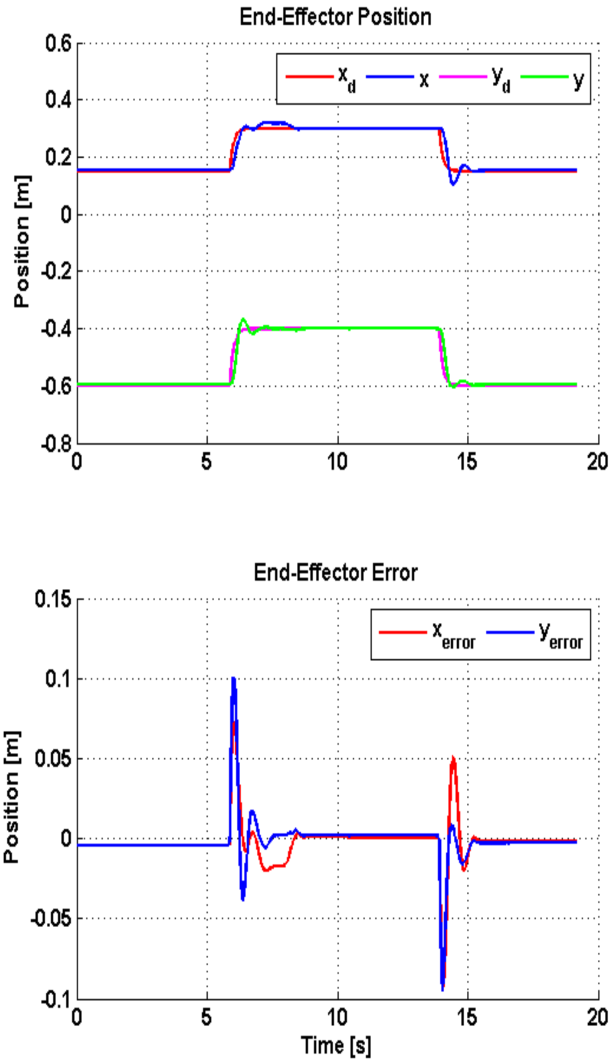


Figure 5. Control Result with Task-Space Torque Controller Based on TDC

offers a more accurate result. It decreases the $x-axis$ error to $0.00005[m]$, and $y-axis$ to $0.004[m]$.

V. Conclusion

In this paper a task-space control method, which is based on virtual spring damper hypothesis, that can eliminate the non-linear friction, and modelling errors was proposed. Advantage of the Time-Delay Control (TDC) method, that does not need the friction and dynamic model, the proposed control method is adaptive in friction existing systems. To prove the controller performance the 3-joint manipulator is used. From the experimental results, the task-space controller has improved it's performance in help of the TDC method.

Acknowledgment

This work was supported by the Global Frontier R&D program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government (MSIP) (2013M3A6A3078404).

References

[1] D. Whitney, "Resolved motion rate control of manipulators and human prostheses,"IEEE Transactions on Man-Machine Systems, vol. 10, no. 2, 1969, pp. 47-53.

[2] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy,"*in Proceedings of Robotics Research: The First International Symposium*, 1984, pp. 735-747.

[3] Y. Nakamura, Advanced robotics: redundancy and optimization, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1990.

[4] A. Albu-Schaffer, S. Haddadin, C. Ott, A.Stemmer, T. Wimbock, and G. Hirzinger, "The DLR Lightweight Robot: Design and Control Concepts for Robots In Human Environments,"Industrial Robot: An International Journal, vol. 34, no. 5, 2007, pp. 376-385.

[5] S. Arimoto, and M. Takegaki, "A New Feedback Method for Dynamic Control of Manipulators," Journal of dynamic systems, measurement, and control, vol. 102, 1981, pp. 119-125.

[6] C. Ott, A. Albu-Schaffer, A. Kugi, and G. Hirzinger, "Decoupling Based Cartesian Impedance Control of Flexible Joint Robots," IEEE International Conference on Robotics & Automation, vol. 3, September 2003, pp. 3101-3107.

[7] S. Arimoto, and M. Sekimoto, "Human-like movements of robotic arms with redundant DOFs: virtual spring-damper hypothesis to tackle the Bernstein problem," IEEE International Conference on Robotics & Automation, May 2006, pp. 1860-1866.

[8] M. Sekimoto, and S. Arimoto, "Experimental Study on Reaching Movements of Robot Arms with Redundant DOFs Based upon Virtual Spring-Damper Hypothesis," IEEE International Conference on Intelligent Robots and Systems, October 2006, pp. 562-567.

[9] S. Arimoto, H. Hashiguchi, M. Sekimoto, and R. Ozawa, "Generation of natural motions for redundant multi-joint systems: A differential-geometric approach based upon the principle of least actions," Journal of Robotic Systems, vol. 22, 2005, pp. 583-605.

[10] S. Arimoto, H. Hashiguchi, M. Sekimoto, and R. Ozawa, "Natural resolution of ill-posedness of inverse kinematics for redundant robots: A challenge to Bernstein's degrees-of-freedom problem," Advanced Robotics, vol. 19, no. 4, 2005, pp. 2484-2495.

[11] S. Arimoto, H. Hashiguchi, and R. Ozawa, "A SIMPLE CONTROL METHOD COPING WITH A KINEMATICALLY ILL-POSED INVERSE PROBLEM OF REDUNDANT ROBOTS: ANALYSIS IN CASE OF A HANDWRITING ROBOT," Asian Journal of Control, vol. 7, no. 2, 2005, pp. 112-123.

[12] J. P. Hauschild, G. Heppler, and J. McPhee, "Friction compensation of harmonic drive actuators," Proceedings of 6th International Conference on Dynamics and Control of Systems and Structures in Space, Jul 2004, pp. 683-692.

[13] T. Murakami, and K. Ohnishi, "Observer-Based Motion Control: Application to Robust Control and Parameter Identification," IEEE Industrial Electronics Society: Asia-Pacific Workshop on Advances in Motion Control, 1993, pp. 1-6.

[14] P. H. Chang, "A Model Reference Observer for Time-Delay Control and Its Application to Robot Trajectory Control," IEEE Transactions On Control Systems Technology, vol. 4, no. 1, January 1996, pp. 2-10.

[15] K. Youcef-Toumi, and O. Ito, "A time delay controller design for system with unknown dynamics," IEEE American Control Conference, vol. 112, no. 1, 1988, pp. 133-142.

[16] T. C. Hsia, and L. S. Gao, "Robot Manipulator Control Using Decentralized Time-Invariant Time-Delayed Controllers," in Proc. IEEE International Conference on Robotics & Automation, 1990, pp. 2070-2075.

[17] S.-K. Kim, J.-H. Bae, and S.-R. Oh, "Concurrent control of position/orientation of a redundant manipulator based on virtual spring-damper hypothesis," IEEE International Conference on Robotics & Automation, 2011, pp. 6045-6050.

[18] S-m. Hur, S.-K. Kim, Y. Oh, and S.-R. Oh, “Joint torque servo of a high friction robot manipulator based on time-delay control with feed-forward friction compensation,” IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2012), September 2012, pp. 37-42.

[19] S-m. Hur, S.-R. Oh, and Y. Oh, “Joint space torque controller based on time-delay control with collision detection,” IEEE International Conference on Intelligent Robots and Systems (IROS 2014), 2014, pp. 4710-4715.

[20] M. Spong, “Modeling and Control of Elastic Joint Robots,” ASME Journal of Dynamics Systems, Measurement and Control, vol. 109, 1987, pp. 310-319.

[21] P. S. Gandhi, “Modeling, Identification, and Compensation of Friction in Harmonic Drives,” Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, Nevada USA, December 2002, pp. 160-166.

[22] H. Olsson, K. J. Astrom, C. Canudas, M. Gafvert, and P. Lischinky, “Friction models and friction compensation,” European Journal of Control, vol. 4, no. 3, 1998.

[23] K. Rafael, and L. Jes*ú*s, “Determination of Viscous and Coulomb Friction by Using Velocity Response to Torque Ramp Inputs,” IEEE International Conference on Robotics & Automation,vol. 3, May 1999, pp. 1740-1745.

[24] B. Xian, M. S de Queiroz, D. Dawson, and I. Walker, “Task-space tracking control of robot manipulators via quaternion feedback,” IEEE International Conference on Robotics & Automation,vol. 20, no. 1, 2004, pp. 160-167.

[25] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, “Task-space control of robot manipulators with null-space compliance,” IEEE Transactions on Robotics,vol. 30, no. 2, 2014, pp. 493-506.

Context-Aware Authentication for the Internet of Things

Kashif Habib, Wolfgang Leister
Norwegian Computing Center
Oslo, Norway
e-mail: Kashif.Sheikh@nr.no, Wolfgang.Leister@nr.no

***Abstract*— Many traditional authentication and access control mechanisms do not use context-aware approach, i.e., those mechanisms do not incorporate context parameters while making authentication and authorisation decisions. The context unaware mechanisms can be inadequate for the Internet of Things due to its dynamic and heterogeneous environment. The context information can be used to reconfigure security mechanisms and adjust security parameters. The contextual information can be integrated into various security mechanisms such as authentication, access control, encryption, etc. The context-aware security is the dynamic adjustment of security policy based on the context. In this paper, we discuss the context-awareness techniques for authentication and access control mechanisms. We present the concepts of context, context- awareness, and context based security and highlight contextual attributes that can be used to support and enhance authentication and access control mechanisms for the Internet of Things.**

Keywords-Context; context-awareness; Internet of Things; authentication; access control; security.

I. Introduction

The Internet of Things (IoT) has gained much popularity in recent past due to integration of smartphones, tablets, and sensor networks into the Internet. The IoT envisions an environment in which sensors, recording devices, smartphones, tablets, and laptops are networked together and are actively monitoring changes in their surroundings. The contextual information can be monitored through various sources, such as sensors deployment, device status, and user's behaviour. The devices collaborate with each other to further facilitate human computer interactions or to provide the environmental information. The data provided by sensors or other recording devices are referred to as contextual data, since they contain information about the context in which each entity or user is located. Mobile devices are one of the common platforms to access resources in the IoT where contextual information can be considered dynamic. The contextual information can be about user's location and behaviour, current time, state of system resources, and state of network and security configurations.

The IoT faces some challenges such as security, privacy, trust, and context-awareness about the surrounding environment and about system state itself. The challenges are important because the IoT is dynamic in nature and does not have very well defined network boundaries. The IoT envisages dynamic and heterogeneous environment in which a context-aware based security can deal with the security problems. The ubiquitous applications can utilise the environmental information for decision making [1]. This means that the security mechanisms developed for the IoT can incorporate the contextual information while making a security decision. A security mechanism can be considered context-aware, if it can spot the event happening in surrounding environment. Context-awareness is an essential element for an authentication system while evaluating associated risks with a system [2].

A. *Motivation and Contribution*

Previously, we have developed an authentication framework based on biometric modalities and wireless device radio fingerprinting [3]. Our framework ensures that the received data at remote medical center belongs to correct patient and identifies the fabricated data. Incorporating context awareness and adaptive security in our framework are challenges because a non-match between stored and given templates always can not be treated as a threat to the system, rather there can be situations where environmental or system's context can assist us in decision making. Adaptive security can make template matching more flexible and we can adjust security level instead of blocking transmission during no-match due to the changed context. In this paper, we discuss and elaborate context, context-awareness, context-aware security, and context-aware authentication concepts for the IoT. While discussing the above mentioned concepts, our main focus is towards context-aware approaches for authentication and access control mechanisms and we classify the mechanisms according to context modelling approaches.

The paper is organized as follow: in Section II, we introduce context and context-awareness concepts. In Section III, we review context-aware security paradigm. The context-aware security models, frameworks, protocols, and prototypes for authentication and access control mechanisms are highlighted in Section IV. Section V contains some discussions and Section VI concludes the paper followed by future work.

II. Context and Context-Awareness

The term context-aware can be defined for different application areas and for different purposes. There are several definitions of context-awareness in the literature [4]. According to Schilit and Theimer [5], "a system is context-aware if it can provide context relevant information and services to users and applications from the set of context types, such as location, identification of nearby people, objects and changes to those objects." Soon after them, Schilit et al. [6] also defined a context-aware system. According to them, “a system is context-aware if it can adapt itself to the context.” Afterwards, many people defined context-aware systems in a similar way. For example, according to Dey [7], “a system can be context-aware if it uses context to provide relevant information and, or services to the user, where relevancy depends on users’ task.” According to Ryan et al. [8], “a system is context-aware if it has the ability to detect and sense, interpret and respond to aspects of a user’s local environment and to the computing devices themselves.” Dey and Abowd [9] define context as “any information that can be used to characterise the situation of an entity that is considered relevant to the interaction between a user and an application”. According to Krish [10] “context is a highly structured amalgam of information, physical and conceptual resources that go beyond the simple facts of who or what is where and when to include the state of digital resources, people concepts and mental state, task state, social relations, and the local work culture, to name a few ingredients”.

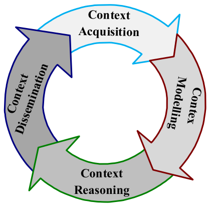


Figure 1. Context life cycle

As depicted in Figure 1, a context-aware system follows the life cycle process to deliver contextual information. Gomez and Wrona [11] identified context information discovery, context information acquisition, and context information reasoning as main steps in a life cycle of context-aware system. Bernardos et al. [12] identified context acquisition, information processing, and reasoning and decision as main phases in a typical context management system. After reviewing the life cycles of context-aware system, Perera et al. [13] derived context acquisition, context modelling, context reasoning, and context dissemination as four phases in a typical context management system.

1) *Context information acquisition:*

A context-aware system collects contextual information from the discovered context information providers and stores it in a context information repository for further reasoning. The context acquisition can also follow pull and push modes. The pull mode allows context-aware system to request contextual information, whereas in case of push mode, context information providers push context information to the context-aware system.

2) *Context information modelling:*

The contextual information is processed in terms of attributes, characteristics, relationships, quality-of context attributes and the queries for synchronous context requests. Afterwards, the new context information is organised and added to the existing contextual information repository for use.

3) *Context information reasoning:*

A reasoning mechanisms facilitate applications to utilise the available context information. In order to establish a reasoning mechanism, a single piece of context information or a collection of such information can be used.

4) *Context information dissemination:*

The applications requiring contextual information use context dissemination to acquire context. The context is disseminated using query and subscription methods. In a query method, the context management system can use that query to produce results. In a subscription method, the applications subscribe the requirements with a context management system that provides the results upon detecting an event.

TABLE I. SUMMARY OF CONTEXT TYPES

Context type	Captured contextual information	Available sensors and technologies
Physical context	Light, temperature, noise, humidity level, traffic conditions.	Photodiodes; biosensors; thermometer; ultraviolet sensors.
Computing context	Network capacity; connectivity; bandwidth; costs of computing and communication; resources such as printers, and workstations; available processors and devices accessible for user input and display.	Touch sensors implemented in mobile devices; microphones; system log; user behaviour monitoring; device log, various environmental sensors.
User context	User location, collection of nearby people, user profiles, social situation.	Active badge system; GPS; camera; mercury switches; GSM; motion detectors; accelerometers.

The three important aspects of context are: where you are, whom you are with, and what resources are nearby [14]. Based on these aspects, context can be divided into three parts: user context, computing context, and physical context. Table I provides a summary of available sensors and technologies to capture contextual information for each context type.

TABLE II. SUMMARY OF CONTEXT ATTRIBUTES

Attributes	Description
Context categories	Conceptual; measurable; static; dynamic; continuous; discrete; internal; external; material; social; physical; virtual; real-time; unreal-time; natural; technology; social; location; identity; time; activity
Context-awareness approaches	*Active context-awareness:* Contextual changes are discovered by detecting changes in the application’s behaviour. *Passive context-awareness:* Applications present the updated context to a user.
Context learning approaches	*Sensed context:* Environment information; user’s physical information; user’s interaction habits and interactive historical records. *Derived context:* Computed on the go; explicit context; user preferences.
Context modelling	Key-value; mark up scheme; graphical; object oriented; logical; ontology.

Table II provides a summary of context attributes. Context is classified according to context categories, context-awareness approaches, context learning approaches, and context modelling approaches.

III. CONTEXT-AWARE SECURITY

Many existing computer networks comply with allow and deny based access control policies. Allow means granting access when the user or device credential matches with pre-stored credentials and deny means blocking access when the user or device credential do not match with pre-stored credentials. This type of system can be considered static in nature because it does not take into consideration other factors such as, contextual information from the user or device environment while making allow and deny decisions. But the IoT has a dynamic environment, where flexible security policies using contextual information can potentially increase the effectiveness of security decisions.

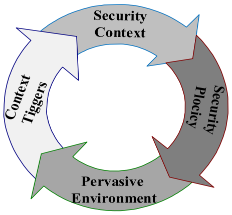


Figure 2. Context-aware security

The security context is defined by Kouadri and Brézillon [15] as: “a set of information collected from the user's environment and the application environment and that is relevant to the security infrastructure of both the user and the application.” Brézillon and Mostéfaoui [16] define the security context as a situation where a security solution considers a set of information while making a specific security decision. For example, while detecting an intrusion during communication, security mechanism may adapt to strong authentication method. As depicted in Figure 2, initially the pervasive computing environment is controlled by some security policy depending upon the initial context at that time [16] [17].

Context triggers refer the dynamic changes in the environment with the passage of time. Security context refers this new context that is to be considered while deploying new security actions as a result of the change. A security policy indicates the rules and regulations that govern who has the access and who doesn’t in each type of situation. Thus, the security policy should be flexible enough to accommodate changing contexts.

Strang and Linnhoff-popien [18] surveyed the relevant approaches to modelling context. The authors reviewed various approaches, classified relative to their core elements, and evaluated with respect to their appropriateness for ubiquitous computing. Many context-aware applications based on various context models have been developed in past for a variety of application domains. The existing approaches to context information modelling are sorted into six categories [18] [19] [20], which are based on the data structures used for lying out and exchanging context data in the respective system. Table III summarises the available security context modelling approaches.

Halunen and Evesti [21] presented some possibilities of utilising context-aware systems in adaptive user authentication settings. They suggested to first use the context information to control an adaptive security system and then linked to the authentication scheme via tags.

TABLE III. SUMMARY OF SECURITY CONTEXT MODELING

Key-value modelling: *Description:* simple key-value pairs to define the list of attributes and their values describing context; information used by context-aware applications. *Strength:* easy to manage; simple data structure to depict the contextual information. *Weakness:* limited capabilities in: (i) capturing a variety of context types; (ii) capturing contextual relationships; (iii) dependencies, timeliness, and quality of context information; (iv) sophisticated structuring for enabling efficient context retrieval algorithms.
Mark-up scheme modelling: *Description:* it uses a variety of mark-up languages; hierarchical data structure consisting of mark-up tags with attributes and content; the content of the mark up tags is usually defined by other mark-up tags. *Strength:* can sort the context information by category, priority, and runtime process. *Weakness:* Limited capabilities in: (i) allowing consistency checking; (ii) supporting reasoning on context, on context uncertainty and on higher context abstractions.
Graphical modelling: *Description:* obtained through transformation algorithms; graph data structures and richer data types, e.g., unified modelling language and object role modelling. *Strength:* generic; hierarchically structured allowing the association of a context with an appropriate action. *Weakness:* lack of support for distributed context model; handling incompleteness; lack of formalism for on line automated access.
Object oriented modelling: *Description:* uses object oriented languages to design the dynamic property of the context; the context information is used as a method applied to an object; context processing details are encapsulated on an object level; access to contextual information is provided through specified interfaces only. *Strength:* favours the trust inside the network; partial validation but often not very formal; reuse can be supported through inheritance and composition. *Weakness:* does not provide the support for interoperability; handling incompleteness; has a flat information model.
Logical modelling: *Description:* the context is defined with facts, predictions or roles; a goal is to form new expressions or facts from previous ones; a logic defines the conditions in which a concluding expression or fact may be derived. *Strength:* formalism; structuring. *Weakness:* uncertainties; time variations; validation issues.
Ontology modelling: *Description:* represents a concept group in a given domain, and the relationship between the different concepts; depicts a domain with a graph of concepts; contextual relationships may be hierarchical or semantic. *Strength:* strong regarding the distributed composition requirement; partial validation is possible; comprehensive set of validation tools available. *Weakness:* uncertainties in handling, scalability issues in searching large data volumes.

The different approaches to model security context possess some weaknesses as mentioned in Table III. For example, key-value modelling based approach possesses weaknesses such as, distributed composition of contextual data, partial validation, information quality, incompleteness, and formalism. Graphical modelling approach also lacks in terms of distributed composition of contextual data. Mark-up scheme and logic based modelling approach can be considered weak in terms of handling incompleteness and ambiguity in contextual data. Object oriented model usually require strong distributed composition requirements which are difficult to manage for the devices in the IoT due to limited resources.

IV. Related Work

The context-awareness for authentication and access control mechanisms has been an active research field among researchers. In this section, we classify the context-aware techniques proposed in the existing literature according to context modelling approaches discussed earlier.

1) *Key-value modelling:*

Hayashi et al. [22] introduced context-aware scalable authentication using multiple passive factors by modulating active factors to authenticate users. The authors proposed a generic probabilistic framework to select appropriate active authentication factors, given a set of passive authentication factors. They developed prototypes, and investigated the feasibility and effectiveness of their proposed framework.

Context-aware mobile biometric authentication based on support vector machines is proposed by Witte et al. [23]. Based on the contextual information measured from the environment, the authors constructed subject-specific context models in order to train support vector machine. The authors demonstrated the feasibility of the proposed architecture by developing a mobile application for data collection purposes.

Said et al. [24] presented a context-aware security controller and proposed to integrate it in the long term evolution/evolved packet system access. The authors motivated the integration of a context-aware security controller to minimize the overall security cost. They showed that the controller activates security mechanisms according to the contextual information such as the application type and the device capabilities.

2) *Mark-up scheme modelling:*

Goel et al. [25] described an authentication framework for a context-aware environment. In order to support role-based and location-based access control, the authors used a combination of a user's context, authentication policies and light weight tagging. The framework has a provision for extension to support other contextual information from available resources, environment, and the users who interact with that environment.

Hu and Weaver [26] presented a dynamic, context-aware security infrastructure for healthcare applications. The access control model extends the role based access control mechanism by associating access permissions with context-related constraints. They described the capability of their model by showing authorization decisions approach based upon context information in addition to roles.

A mechanism for modelling complex and interwoven sets of context-information by extending ambient calculus with new constructs and capabilities is presented by Kjægaard and Pedersen [27]. According to the authors, the calculus is a step in the direction of making formal methods applicable in the area of pervasive computing. In particular, the authors identified the key area of the expressiveness of formal models of context-awareness which are represented as hierarchical and independent sets of information.

3) *Graphical modelling:*

Feng et al. [28] incorporated contextual information to improve user authentication by presenting a touch based identity protection service. In order to authenticate a user on continuous basis, they analysed real life touch data as well as underlying contextual information.

Lenzini [29] presented trust-based and context-aware authentication in a software architecture for context and proximity-aware services. The author described context management architecture for context-aware services. The software based architecture collects, arranges, and elaborates high-level contextual information from a sensor network. The author used contextual information to distinguish among different identities, and to evaluate to which extent they are authentic.

Bandinelli et al. [30] presented a context-aware security framework for next generation mobile networks. The authors introduced a context-aware security framework for addressing the problems of end-to-end security on behalf of end-users in a next generation network scenario. Their security framework uses contextual graphs to define security policies encompassing actions at different layers of communication systems while adapting to changing context.

4) *Object oriented modelling:*

Badram et al. [31] presented context-aware user authentication, supporting proximity-based login in pervasive computing environment. The authors introduced a concept of proximity-based user authentication in a pervasive computing environment. User identification is performed through a Java smart card and a context-aware system.

5) *Logical modelling:*

Zhang et al. [32] presented the context-aware access control model for pervasive applications using dynamic role based access control scheme. Based on the context information, the operation of the model extends the role based access control model to dynamically adjust the role assignments, and permission assignments. However, their access control scheme may not be sufficient alone until it is combined with feasible authentication mechanisms to secure pervasive applications.

To improve existing network security protocols in an Intranet environment, Wullems et al. [33] introduced context-aware authorization architecture. The proposed architecture is an extension to role based access control mechanism facilitating context-aware access control policy. They described the implementation of the architecture using dynamic context services and also presented the description of an application utilising their proposed architecture.

An adaptive access control model for medical data in body and wireless area network is designed and developed by Maw et al. [34]. They evaluated the framework using medical scenario in which they included a user behaviour trust module along with the access control module. They concluded that the overriding policy is useful to handle unanticipated situations and showed that by incorporating user behaviour into access control model, one can make better security decisions.

Malek et al. [35] presented a framework for context-aware authentication services in context-aware computing environments. The proposed framework is capable of enabling the users to take initiatives in the context-aware computing environments depending on their desired confidence level. To establish trust and to share secrets between parties, the context-aware authentication service uses context-data.

Hulsebosch et al. [36] described the theoretical background for a context-sensitive adaptation of authentication. The authors designed and validated the system to adaptively authenticate a user on the basis of the location of his sensed identity tokens. The authors argued that authentication and access control can be made less intrusive, more intelligent, and able to adapt to the rapidly changing contexts of the environment.

Brosso et al. [37] presented a continuous authentication system based on user behaviour analysis. The system utilises environmental context information, user' behaviour analysis, and neuro-fuzzy logic. The authors verified the system with tests and simulations to authenticate a person's identity using behaviour analysis and trust restriction. They used contextual information to establish evidence of user behaviour. The trust levels were decided based upon user behaviour.

6) Ontology modelling:

To provide a security framework suitable for people with disabilities, Mhamed et al. [38] suggested using various contextual data monitored through sensors. The approach shows how to model trust and access control based on user behaviour and capabilities that can be extracted from the monitored data through sensors. The proposed access control model is based on the semantic web technologies.

Wrona and Gomez [11] investigated different aspects of security related to context information. According to them, security challenges in context-aware systems include integrity, confidentiality, availability of context information, and end user's privacy. Trustworthiness of context information is also an important element, which a context information requester can put in the delivered context information.

V. Discussion

Understanding the contextual information is an important element for the IoT. A context-aware system can be considered different from traditional systems because of their capabilities to capture and incorporate environmental factors into decision making process. Particularly, in case of the IoT where device and user attributes such as, location, time, and behaviour can change rapidly, it may be very important for security mechanisms to react based on the changing parameters and adapt accordingly.

Authentication and access control are important security services for the IoT that are needed to check the identity of users and to decide which resources they can access to. The existing authentication mechanisms that are developed for traditional computer network environments are mostly context unaware, and usually do not incorporate contextual information while authenticating a user and a device. But due to dynamic environment and changed context, the threat profile can vary and static authentication mechanism may not be sufficient enough to continue securing a system. Contextual information can help authentication system to know user state and make better identification decisions. The strength of an authentication mechanism can be improved if we broaden our authentication scope beyond the identification of user credentials. Rather, if we can also incorporate the context information, such as user location, user state, and surrounding environmental state, along with user credentials.

While adding context into authentication and access control mechanisms, sometimes incomplete or imprecise context can lead to false positives and false negatives. For example, user and environmental context may be inaccurately determined or context determination may be affected by environmental conditions, etc. Thus, if context acquisition is performed wrongly, it can possibly generate false positives and false negatives. However, if context acquisition and reasoning are performed correctly, and proper context composition techniques are used, then adding context into security decision can reduce the rate of false positives.

VI. Conclusion and future work

Although, developing authentication and access control mechanisms has been an active research areas among researchers, but mostly the existing mechanisms work on the principles of user credential based approach. Context-awareness has a tendency to enhance the effectiveness of those mechanisms by incorporating contextual data into a decision making process. In this paper, we highlighted the necessary concepts of context, context awareness, and context based security. In addition, the approaches proposed in the existing literature, regarding incorporating context-awareness into authentication and access control mechanisms in the IoT are presented.

Previously, we have developed an authentication framework based on biometric and radio fingerprinting for the IoT in eHealth. In future, the work in this paper will be used as a basis for the development of context-aware authentication mechanisms for the IoT in eHealth. Precisely, we will carry out context-awareness modelling for our earlier developed framework.

Acknowledgment

The work presented here has been carried out in the research project ASSET – Adaptive Security for Smart Internet of Things in eHealth (2012–2015) funded by The Research Council of Norway.

References

[1] A-C. Pierre, "A dynamic trust-based context-aware secure authentication framework for pervasive computing environments," PhD thesis, Computer Science, Institut National des Télécommunications, France, 2010.

[2] RSA Risk-based Authentication, white paper, November 2013, pp. 1-4.

[3] K. Habib, A. Torjusen, and W. Leister, “A Novel Authentication Framework Based on Biometric and Radio Fingerprinting for the IoT in eHealth,” In proceedings of SMART 2014, July 20 - 24, 2014, pp. 32-37.

[4] S. Poslad, “Ubiquitous Computing: Smart Devices, Environments and Interactions,” Wiley Publishing, 2009.

[5] B. Schilit and M. Theimer, “Disseminating active map information to mobile hosts,” IEEE Networks, 8(5): , pp. 22-32, 1994.

[6] B. Schilit, N. Adams, and R. Want, “Context aware computing applications,” In Proceeding of 1st International Workshop on Mobile Computing Systems and Applications, 1995, pp. 85-90.

[7] A. K. Dey, “Providing Architectural Support for Building Context Aware Applications,” PhD thesis, Computer Science, Georgia Institute of Technology, Atlanta, November, 2000.

[8] N. Ryan J. Pascoe, and D. Morse, “Enhanced reality fieldwork: the context aware archaeological assistant,” In V. Gaffney, M. van Leusen, and S. Exxon (eds) Computer Applications in Archaeology, British Archaeological Reports, 1997.

[9] A. K. Dey and G. D. Abowd, “Towards a better understanding of context and context awareness,” In Proceedings of the (HUC '99), 1999, pp. 304-307.

[10] D. Kirsh, “The Context of Work, Human-Computer Interaction,” vol. 16, 2001, pp. 305-322.

[11] K. Wrona and L. Gomez, “Context-aware security and secure context-awareness in ubiquitous computing environments,” XXI Autumn Meeting of Polish Information Processing Society, 2004, pp.255-265.

[12] A. Bernardos, P. Tarrio, and J. Casar, “A data fusion framework for context-aware mobile services,” IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2008, pp. 606 –613.

[13] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” IEEE, 16, 2014, pp. 414-454.

[14] W. Liu X. Li, and D. Huang, “A survey on context awareness,” International Conference on Computer Science and Service System (CSSS), 27-29 June 2011, pp. 144-147.

[15] G. K. Mostéfaoui and P. Brézillon, “Modeling Context-Based Security Policies with Contextual Graphs,” In Proceedings of (PERCOMW '04). IEEE Computer Society, 2004, pp. 28-32.

[16] P. Brezillon and G. K. Mostefaoui, “Context-based security policies: a new modelling approach,” Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, March 2004, pp.154,158.

[17] P. Shetty and S. W. Loke, “Context-Based Security (and Safety) Meta-Policies for Pervasive Computing Environments: the case of Smart Homes,” In the Workshop on Context and Safety at Context, 2005, pp. 1-14.

[18] T. Strang and C. Linnhoff-Popien, “A Context Modelling Survey,” In Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004, pp. 1-8.

[19] S. Vuppala et al., “uBiquitous, secUre inTernet-of-things with Location and contEx-awaReness,” BUTLER project, D2.1 - Requirements, Specifications and Security Technologies for IoT Context-Aware Networks, October 2012, pp. 1-171.

[20] C. Bettini et al., “A survey of context modelling and reasoning techniques,” Pervasive Mob. Comput. 6, (2), April 2010, pp. 161-180,.

[21] K. Halunen and A. Evesti, “Context-Aware Systems and Adaptive User Authentication,” Evolving Ambient Intelligence, Springer International publishing, 413, 2013, pp.240-251.

[22] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley, “CASA: context-aware scalable authentication,” In Proceedings of (SOUPS '13). ACM, Article 3, 2013, pp. 1-10.

[23] H. Witte, C. Rathgeb, and C. Busch, “Context-Aware Mobile Biometric Authentication based on Support Vector Machines," In proceedings of the (EST), 2013, pp.29-32.

[24] S. B. H. Said, K. Guillouard, and J-M. Bonnin, “On the benefit of context-awareness for security mechanisms in LTE/EPS networks,” In proceedings of the (PIMRC), 2013, pp. 2414-2428.

[25] D. Goel, E. Kher, S. Joag, V. Mujumdar, M. Griss, and A. Dey, “Context-Aware Authentication Framework Mobile Computing, Applications, and Services,” Springer Berlin Heidelberg, 35, 2010, pp. 26-41.

[26] J. Hu and A. C. Weaver, “A Dynamic, Context-Aware Security Infrastructure for Distributed Healthcare Applications,” In Proceedings of the (PSPT), 2004, pp. 1-8.

[27] M. B. Kjægaard and J. Bunde-Pedersen, “Towards a Formal Model of Context Awareness,” In proceedings of the (CTSB), 2006, pp. 1-8.

[28] T. Feng, J. Yang, Z. Yan, E. Tapia, and W. Shi, “TIPS: context-aware implicit user identification using touch screen in uncontrolled environments,” In Proceedings of the (HotMobile '14), ACM, Article 9, 2014, pp. 1-6.

[29] G. Lenzini, “Trust-Based and Context-Aware Authentication in a Software Architecture for Context and Proximity-Aware Services Architecting Dependable Systems” VI, Springer Berlin Heidelberg, 5825, 2009, pp. 284-307.

[30] M. Bandinelli, F. Paganelli, G. Vannuccini, and D. Giuli, “A Context-Aware Security Framework for Next Generation Mobile Networks Security and Privacy in Mobile Information and Communication Systems,” Springer Berlin Heidelberg, 17, 2009, pp. 134-147.

[31] J. Bardram, R. Kjær, and M. Pedersen, “Context-Aware User Authentication- Supporting Proximity-Based Login in Pervasive Computing,” Ubiquitous Computing, Springer Berlin Heidelberg, 2864, 2003, pp.107-123.

[32] G. Zhang and M. Parashar, “Context-aware dynamic access control for pervasive applications,” Proceedings of the Communication Networks and Distributed Systems Modelling and Simulation Conference, 2004, pp. 21-30.

[33] C. Wullems, M. Looi, and A. Clark, “Towards context-aware security: an authorization architecture for intranet environments,” In proceedings of the (PERCOMW’04), March 2004, pp.132-137.

[34] H. A. Maw, H. Xiao, and B. Christianson, “An adaptive access control model for medical data in Wireless Sensor Networks," IEEE 15th International Conference on e-Health Networking, Applications & Services, 2013, pp.303-309.

[35] B. Malek, A. Miri, and A. Karmouch, “A framework for context-aware authentication," 4th International Conference on Intelligent Environments, IET , 21-22 July 2008, pp.1-8.

[36] R. Hulsebosch, M. Bargh, G. Lenzini, P. Ebben, and S. Iacob, “Context Sensitive Adaptive Authentication,” smart sensing and context,” Springer Berlin Heidelberg, 4793, 2007, pp. 93-109.

[37] I. Brosso, A. La Neve, G. Bressan, and W. V. Ruggiero, “A Continuous Authentication System Based on User Behavior Analysis,” In proceedings of the (ARES '10), Feb. 2010, pp. 380-385.

[38] A. Mhamed, M. Zerkouk, A. El Husseini, B. Messabih, and B. El Hassan, “Towards a Context Aware Modelling of Trust and Access Control Based on the User Behaviour and Capabilities Inclusive Society, Health and Wellbeing in the Community, and Care at Home,” Springer Berlin Heidelberg, 7910, 2013, pp. 69-76.