ADAPTIVE 2012

The Fourth International Conference on Adaptive and Self-Adaptive Systems and Applications

ISBN: 978-1-61208-219-6

July 22-27, 2012

Nice, France

ADAPTIVE 2012 Editors

Dana Petcu, West University of Timisoara, Romania

Elena Troubitsyna, Åbo Akademi University Finland

ADAPTIVE 2012

Foreword

The Fourth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2012), held between July 22 and 27, 2012 in Nice, France, targeted advanced system and application design paradigms driven by adaptiveness and self-adaptiveness. With the current tendencies in developing and deploying complex systems, and under the continuous changes of system and application requirements, adaptation is a key feature. Speed and scalability of changes require self-adaptation for special cases. How to build systems to be easily adaptive and self-adaptive, what constraints and what mechanisms must be used, and how to evaluate a stable state in such systems are challenging duties. Context-aware and user-aware are major situations where environment and user feedback is considered for further adaptation.

We take here the opportunity to warmly thank all the members of the ADAPTIVE 2012 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ADAPTIVE 2012. 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 are grateful to the members of the ADAPTIVE 2012 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ADAPTIVE 2012 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of adaptive and self-adaptive systems and applications.

We are convinced that the participants found the event useful and communications very open. We hope Côte d'Azur provided a pleasant environment during the conference and everyone saved some time for exploring the Mediterranean Coast.

ADAPTIVE 2012 Chairs:

ADAPTIVE Advisory Chairs
Radu Calinescu, Aston University, UK
Thomas H. Morris, Mississippi State University, USA
Serge Kernbach, University of Stuttgart, Germany
Antonio Bucchiarone, FBK-IRST of Trento, Italy

ADAPTIVE 2012 Industry/Research Chairs

Dalimír Orfánus, ABB Corporate Research Center, Norway
Weirong Jiang, Juniper Networks Inc. - Sunnyvale, USA

ADAPTIVE 2012

Committee

ADAPTIVE Advisory Chairs

Radu Calinescu, Aston University, UK
Thomas H. Morris, Mississippi State University, USA
Serge Kernbach, University of Stuttgart, Germany
Antonio Bucchiarone, FBK-IRST of Trento, Italy

ADAPTIVE 2012 Industry/Research Chairs

Dalimír Orfánus, ABB Corporate Research Center, Norway
Weirong Jiang, Juniper Networks Inc. - Sunnyvale, USA

ADAPTIVE 2012 Technical Program Committee

Sherif Abdelwahed, Mississippi State University, USA
Habtamu Abie, Norwegian Computing Center/Norsk Regnesentral-Blindern, Norway
Muhammad Tanvir Afzal, Mohammad Ali Jinnah University- Islamabad, Pakistan
Jose Maria Alcaraz Calero, Hewlett-Packard Laboratories-Bristol, UK
Giner Alor Hernández, Instituto Tecnológico de Orizaba - Veracruz, México
Richard Anthony, University of Greenwich, UK
Flavien Balbo, Université Paris-Dauphine, Lamsade-CNRS, France
Luciano Baresi, Politecnico di Milano, Italy
Imen Ben Lahmar, Institut Telecom SudParis, France
Yuriy Brun, University of Washington - Seattle, USA
Radu Calinescu, Aston University, UK
Aldo Campi, University of Bologna, Italy
Valérie Camps, IRIT-Toulouse, France
Bogdan Alexandru Caprarescu, West University of Timisoara, Romania
Radu Calinescu, University of York, UK
Chris Cannings, University of Sheffield, UK
Carlos Carrascosa, Universidad Politécnica de Valencia, Spain
Po-Hsun Cheng, National Kaohsiung Normal University, Taiwan
José Alfredo F. Costa, Federal University, UFRN, Brazil
Carlos E. Cuesta, Rey Juan Carlos University, Spain
Heiko Desruelle, Ghent University - IBBT, Belgium
Mihaela Dinsoreanu, Technical University of Cluj-Napoca, Romania
Ioanna Dionysiou, University of Nicosia, Cyprus
Shlomi Dolev, Ben Gurion University, Israel
Bruce Edmonds, Manchester Metropolitan University, UK
Alois Ferscha, Johannes Kepler Universität Linz, Austria
Ziny Flikop, Consultant, USA

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

Personalized Multimedia Content Generation Using the QoE Metrics in Distance Learning Systems

Aleksandar Karadimce
University for information science and technology
"St. Paul the Apostle"
Ohrid, R. Macedonia
aleksandar.karadimce@uist.edu.mk

Danco Davcev
University for information science and technology
"St. Paul the Apostle"
Ohrid, R. Macedonia
dancho.davchev@uist.edu.mk

***Abstract*—Personalization of multimedia learning content means to classify the multimedia content to meet a specific user's individual interest, preference, background and situational context – captured by a user profile. Appropriate choice of multimedia learning content has become as one of the most important challenges in order to provide user with personalized distance learning material. The main contribution of this paper is the creation of new model for adaptive multimedia learning that dynamically changes the content depending of the mapping relations between the QoS and QoE metrics. The proposed model delivers personalized multimedia content tailored to user cognitive style and adapting the content according the context – aware network conditions. Main focus is given to tracking the context-aware user behavior in order to generate an appropriate and fine-grained user profile with personalized learning content. The simulation of different models of individual user profiles were developed using the OPNET network simulation software. The presented simulation results confirmed the correctness of the developed model.**

Keywords-personalized multimedia content; adaptive distance learning system; context-aware; OPNET; user profile.

I. INTRODUCTION

Current distance learning systems provide learning content that can be reused and easily shared regardless of the location of the student, anywhere and anytime. In this way, more emphasis is given to the process of indexing and manipulating of learning content in order to create multiple smaller packages called learning assets [1]. These assets are collected in a SCORM (Shareable Content Object Reference Model) compliant repository in order to create and deliver the learning materials [2]. Using the modern Web 2.0 technologies, as in [3], easy delivery of learning materials and increased interoperability among various e-learning platforms has been provided. In this way, reusable SCORM learning objects can be very easy imported and exchanged between various e-learning platforms if they are SCORM-compliant [9].

In the existing learning process, teachers handover their knowledge according to predefined learning schedule and do evaluation of student knowledge using different kinds of tests, quizzes and questionnaires. Therefore, it is imperative to create customized interactive process of knowledge transfer in such a way that students receive tailored course according to their reasoning preferences. In this process, it is important that learning material does not force them to split their attention between multiple sources of information. This way, combination of different media type can provide appropriate personalized learning environment. This can be considered as necessity to implement the multimedia learning, in particular that a mix of different media enhances learning experience [14].

Separation of cognitive learning styles using the visual and verbal perception has proven to be successful for different learner groups: undergraduate, postgraduate and as well educators in higher education. These groups prefer more visual perception than the typically available existing e-learning environments. The visual material includes images, graphics, simulations and videos in order to increase overall quality of experience for the learner. On the other hand, this does not imply however to neglect the verbal perception that consist of audio and text-based learning [14].

Developing a model for adaptive multimedia learning that will provide dynamic adaptation of a network's operation and performance (for example bandwidth) should be in line with user's profile dynamic requests. In this way, the multimedia content will be adapted according to available network resources and the user learning profile. Usually, environment is greatly influenced by user perception of quality that is measured by the QoE (Quality of Experience) metrics as a multi-dimensional construct of user perceptions and behaviors. In our proposed model, the QoE metrics is used for estimating cognitive learning styles of the end user, as visualizer, verbalizer or bimodal perception of information. If we consider the technological systems as "environments", their influence is quantified by QoS (Quality of Service) metrics. They lead to subjective and objective responses of users, both of which are considered as part of the "user experience" [5].

This paper is structured as follows. Section II describes the proposed context-aware multimedia learning model and discusses how QoE is perceived in current research. Section III presents our proposed scheme of mapping between QoS and QoE in order to deliver personalized learning content. In section IV, we will present our simulation results. Section V gives the conclusion and future work.

 ISBN: 978-1-61208-219-6

II. CONTEXT-AWARE MULTIMEDIA LEARNING MODEL

In order to create a personalized multimedia learning content, we need to shift towards estimating the QoE. That is more associated to a specific user profile and the degree of user satisfaction from a given multimedia service or product. The International Telecommunication Union defines QoE as: "The overall acceptability of an application or service, as perceived subjectively by the end-user" [6]. Because QoE is something that is created by user's observance, it is more a qualitative measure than a quantitative one. Then, it needs comprehensive subjective quality assessment methodologies and objective quality assessment metrics that will model the human perception as closely as possible.

On the other hand, QoE offers a different way of measuring how users observe the existing applications. This measurement provides means how to measure overall user satisfaction of a service and hence, to enhance and adjust the existing multimedia delivery content when needed. In this way, in order to maintain a satisfactory QoE, certain threshold for QoS metrics during a communication session is needed. The adaptive mechanism has to provide personalized learning materials in real-time according to the context (user's knowledge and environment) and users previous feedback records. Therefore, we are encouraged to develop a QoE description method based on existing network parameters that are measured by QoS metrics in order to estimate user's profile. This way user profile stores learner's personal level of knowledge, cognitive style and current network conditions.

The area of distance IP-oriented learning, known also as e-learning, is growing very fast and rapidly. There has been different adaptive e-Learning Systems developed in order to support learning style based to user's level of knowledge, using the AHA (The Adaptive Hypermedia Architecture) [7] and Adaptive educational hypermedia systems [8]. The approach for adaptation in technology driven learning has significantly improved learning process by adapting course content presentation to user learning styles and has been implemented in the AEHS-LS system (Adaptive E-Learning Hypermedia System based on Learning Styles) [12].

In the research area for creating personalized multimedia content one of the existing solutions propose MM4U ("multimedia for you") software framework to support the dynamic creation of personalized multimedia content [16]. In the proposed generic and modular framework the authors need to choose, using authoring tool, the media elements that are suitable for the intended user. We have advanced this procedure of authoring in the new model for adaptive multimedia learning with the introduction of user profiling.

This was a motive to focus our research in direction to improve the process of learning by analyzing the impact of user *cognitive* style for the provided multimedia content type. Human perception in the learning process is enhanced by delivery of diversity of reach multimedia content. Many dimensions of cognitive learning styles are offered to the users, among which focus is given to Visualizer/Verbalizer presentation of information. The differences between Visualizers and Verbalizers are frequently not as great as some other cognitive styles. Indeed many bimodal users are equally comfortable using either modality [4]. Visualizers prefer to receive multimedia information via graphics, animation, video and images, whereas, mainly because their visual memory is much stronger than their verbal. On the other hand, verbalizers would prefer to process information in the form of words, expressed by audio or text based form.

Creating personalized multimedia content with the new model for adaptive multimedia learning and the context in which this content is delivered was taken into consideration. Besides it has also influenced by the network conditions. Therefore, the multimedia content has to be dynamically changed with the network conditions, in order to guarantee the quality of service parameters. On the other hand, we have the user cognitive perception of multimedia content that offers diversity in selection of the type of multimedia content. In addition, the context-aware transfer needs to be taken into consideration. Creating multimedia content that is personalized for a certain user profile will need appropriate weighted factors for each of the cognitive learning styles and appropriate context- aware factors. In order to ensure a better user experience for the delivered multimedia content, we propose the content to be network-aware and the network to be context- aware.

Process of creating personalized user profile includes gathering cognitive learning styles of the user that are later used to create dynamic content multimedia delivery. Modeling adaptive learning content that is dynamically generated from the available content is done according to the user profile. The estimation and preferences of users profile

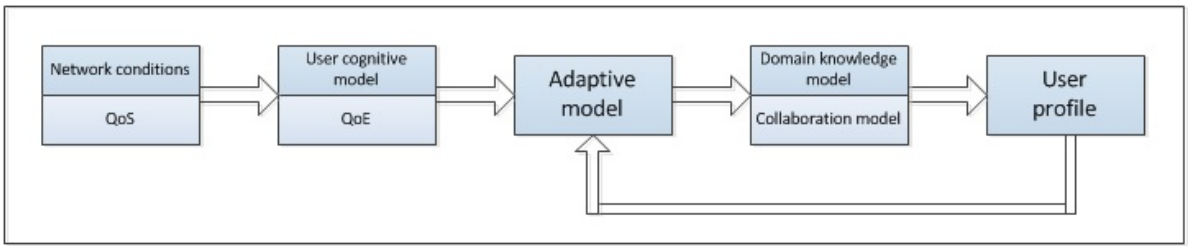


Figure 1. Model of adaptable dynamic multimedia context- aware (ADMCA) distance learning system

are modeled by assessment of its level of knowledge and learning style. In this way, we are able to personalize the multimedia learning course and provide content that is close to its cognitive reasoning preferences. QoE metrics is used for estimating cognitive learning style as visualizer, verbalizer or bimodal perception of information. This dimension is mapped with the context aware network delivery condition, measured by QoS in order to deliver adaptable multimedia content to the users.

In Figure 1, we present our Adaptable Dynamic Multimedia Context- Aware (ADMCA) distance learning model. Our model provides optimization of users profile by dynamically adapting the content, based on both user cognitive styles and current network delivery conditions. The Domain knowledge model is designed to store the level of educational content, being organized in a hierarchical structure of concepts, amongst which logical relationships exist. The model provides improved user multimedia interaction that is expressed by the Collaboration model. Role of the Adaptation model is to decide on personalization and adaptations of the multimedia content, based on the information feedback by the Domain knowledge model and Collaboration model. This architecture ensures an adaptive model for dynamically changing the appearance of media content that has been delivered to user. This will allow them to interact and automatically derive user profile from those interactions in order to dynamically update the resulting presentation according to these preferences.

The delivery of high quality personalized leaning content that is SCORM compliant is consisted from more small and reusable modules that can be combined with each other in order to create personalized multimedia content. This model is SCORM compliant, which is fundamental reference model for developing models of learning content and delivery [9]. The proposed ADMCA model for distance learning system extends the current SCORM framework and generates dynamic reusable multimedia learning content based on different cognitive preferences of the users, domain knowledge and collaboration model. This model provides real-time dynamic change of the multimedia content according to the user cognitive model and the current network conditions. Appropriate use of the bandwidth requirements proficiently is needed to make assessment of the current network delivery conditions. This assumes to provide adaptive intelligent user interfaces addressing diverse capabilities of device classes and an automatic adjustment of content regarding the measured QoS parameters.

The results produced by the simulations in this paper provide justification of the model that can be implemented in diversity of education areas. Demand starts from education in primary schools, through the special medical schools (such as autistic, Deaf & Hard of Hearing, etc.) to universities. The proposed model for adaptive multimedia learning that dynamically changes the content, is innovative because analyses the dependences of the mapping relations between the QoS and QoE metrics.

III. MAPPING THE RELATIONS BETWEEN QOS AND QOE

In the newly proposed model for adaptive multimedia learning that creates user personalized multimedia content, we explore the methodology of QoS to QoE metrics mapping and their correlation. The quality estimation part of the model is implemented with a mapping table between QoS and QoE metrics. Table 1 shows the mapping of the context-aware network QoS bandwidth to QoE - Cognitive learning styles in process of generating dynamic multimedia content in learning lesson. Focusing to the main research problem, profiling of system users is done by tracking the context – aware user behavior.

In order to adapt the multimedia content, firstly the context-aware bandwidth needs to be considered in the appropriate user profile. The increased network usage will result in poor available bandwidth and vice versa. We have assumed that the adaptation only considers available bandwidth as it is the parameter affecting most other network parameters as well.

In this context, other aspects need to be considered for improving the learners experience except the need of adaptation to available bandwidth. We go further and take into consideration the user cognitive perception. In this way, in order to generate dynamic multimedia content that will increase the experience that is not only depended from the QoS, we need to express the user perception of quality measured by QoE.

Users should prefer verbal information to be accompanied by visual information and it is more effective to present the verbal information as audio content rather than give an on-screen text. Learning is also improved if animation and audio content are combined in one source and therefore presented simultaneously [14].

TABLE I. MAPPING BETWEEN QOS AND QOE

QoS - Bandwidth	**QoE - Cognitive learning styles**		
	Verbalizer	***Bimodal***	***Visualizer***
good	HQ[1] dynamic Animation, Text with audio	HD[2] video with audio, Simulation, Text with audio	HD[2] video, 3D graphics, HQ images, Dynamic animations
medium	Images, Video with audio, more text description	Video with audio, Images/ 2D graphics, text	Animations, Images / 2D graphics, text
low	audio, more text description	Images, text	Images, icons, text

1. HQ - *High Quality*
2. HD - *High-Definition*

Classification of the multimedia type is done into 2 groups, static media, such as texts, images, graphics, and dynamic media, including audio and video data. In this way, according to the context preferences, the system is doing

adaptive change of the multimedia type. The assessment of the network conditions, expressed by the bandwidth is classified as low, medium or good. This classification is needed in order to adjust specific types of multimedia content, such as audio and video files, that require more bandwidth [13], that vary considerably in the context of the available bandwidth.

In an environment that has *poor bandwidth* the multimedia content that can be streamed is limited to a static media (text- based, images, icons) and audio samples. Other aspects should be also taken into account such as the user cognitive learning styles that limit the verbalizer users to have only text- based and audio learning assets. For visualizers, the enrichment is done in the multimedia lessons with simple images and icons to the text - based learning. The network that has *medium bandwidth* quality will only offer graphics and sample animations in the process of dynamic multimedia content creation. In this environment, verbalizers are introduced with **video and audio media** that is more important for them, than the images and animation learning content. On the other hand, the visualizers are provided with **animations and graphics** in the existing multimedia content. Bimodal users have access to this multimedia content: video with audio, images/graphics and text. Network with *good bandwidth* has opportunity to use: HQ – dynamic Animations, HD video with audio, 3D graphics and HQ images in order to adopt the multimedia content with high quality. To verify the correctness of the developed model we have compared the simulation results from three different scenarios that demonstrate low, medium and high load bandwidth environments.

IV. SIMULATION RESULTS

Multimedia content in particular, including audio and video files, consume a lot of network resources such as bandwidth and this pose stringent requirements on performance parameter levels such as e.g., load and delay [11]. We have used the clustering process of classifying the learning assets into user profiles that are significant in the context of network conditions. The classification has been done into 3 groups of user profiles: Verbalizer, Visualizer and Bimodal user profiles.

The simulation models of individual user profiles were developed using the OPNET network simulation software that provides virtual network communication environment. OPNET provides a comprehensive development environment with a full set of tools including model design, simulation, data collection, data analysis and supporting the modeling of communication networks [10]. Hardware and software resources used for running the OPNET simulations are as follows:

- Workstation ASUS, with CPU Intel Core i5-480M, 2.66 GHz that has Memory of 2 GB RAM
- Microsoft Windows XP® Service Pack 2 (SP2) and Microsoft Visual Studio 2008

This environment provides a way to model the network behaviors by calculating the interactions between modeling devices. We have used the Discrete Event Simulation (DES) because enables modeling in a more accurate and realistic way. It creates an extremely detailed, packet-by-packet model for the activities of network to be predicted [15]. The network simulator was configured to run 1 hour distance learning course.

To model an application in OPNET, the application definition is used to specify/choose the required multimedia types among the various available applications such as text, video conferencing, voice, images and animations. Profile definition was used to create custom user profiles, classified as Verbalizer, Visualizer and Bimodal profiles. These profiles where specified on three different client nodes: Student1, Student2 and Student3 in three different network simulation scenarios (see Figure 2). The Application server (APP_Server) was configured to be able to support all of the applications based on the predefined user profiles.

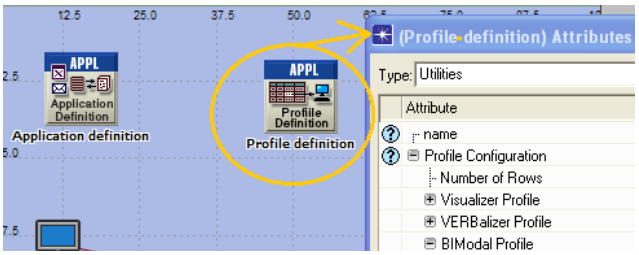


Figure 2. Profile definition model

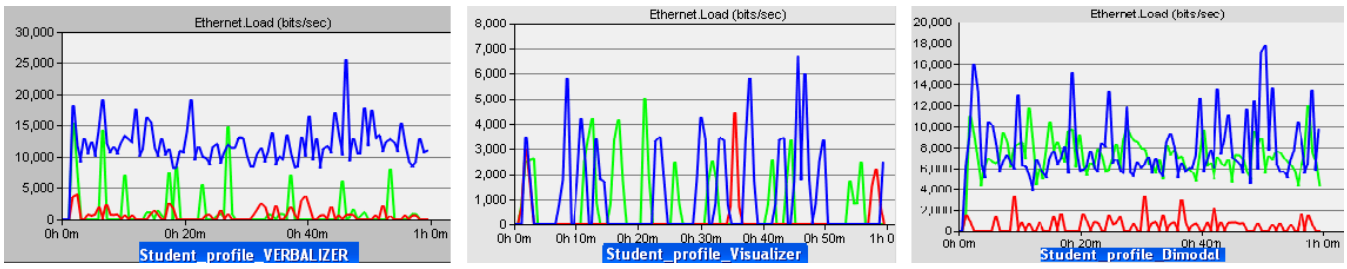


Figure 3. Verbalizer, Visualizer and Bimodal user profiles in different load scenario

Statistics that were collected from the whole network model during a discrete event simulation and the object statistics were collected from the nodes. For our analysis we have chosen object statistics of Ethernet that include delay, throughput and load in analyzing the performance of the proposed scenarios.

After running the simulation for the three different scenarios, with low, medium and high load bandwidth in OPNET simulator, we have done comparison of the traffic load. Represented by the blue line is the High Load scenario, green line shows the results from the Medium load scenario and red line is scenario with Low load.

From the results in Figure 3, we can conclude that in all defined Verbalizer, Visualizer and Bimodal user profiles, the high load consumes the biggest part of the bandwidth that is available. Also, the same results confirm that the three user profiles took the lowest part of the bandwidth during the Low load scenarios which confirms the established mapping between QoS and QoE. The Medium load scenario has a balanced distribution part of the bandwidth in all three user profiles.

Comparison of the throughput between user profiles in three different network simulation scenarios is given in Figure 4. Results confirmed that the network simulation scenario with high load carries the biggest throughput for all three user profiles.

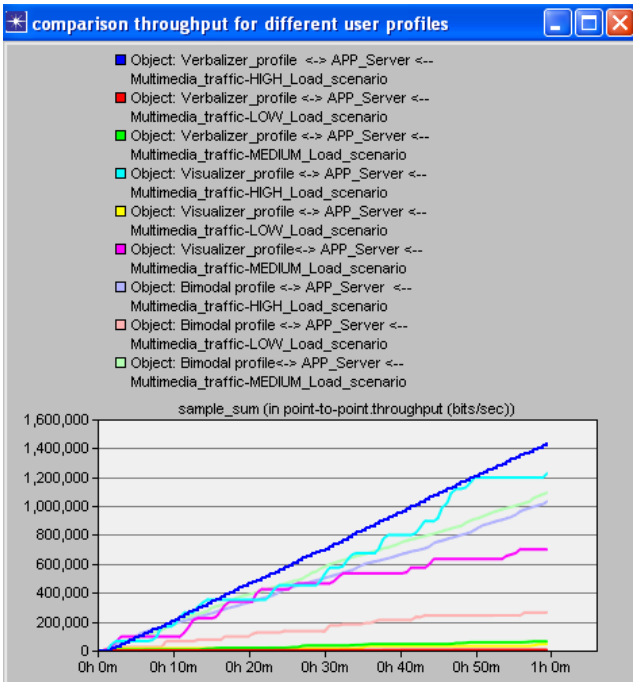


Figure 4. Comparison of the throughput between the user profiles

Results from average network delay comparison for three different network simulation scenarios are given in Figure 5. They confirmed that in Low load scenario we have biggest delay during the transfer of multimedia content that is within the expected conditions. However it is highly important in the high load scenario that the average network delay remains at low level in spite of the increase of number of simultaneous multimedia users.

V. CONCLUSION AND FUTURE WORK

We have developed new model for adaptive multimedia learning that has customized interactive process of multimedia transfer in a context- aware adaptation model that creates dynamic multimedia content. The proposed model is based on the QoE metrics, defined as user cognitive styles. Context-aware adaptation of network conditions should be in line with user cognitive styles in order to offer dynamic adaptation of the multimedia type. These conditions in the newly proposed model have been confirmed by mapping of the context-aware network QoS (in our case the bandwidth) to QoE - cognitive learning style as verbalizer, visualizer or bimodal conditions. This domain independent model supports the creation of personalized multimedia content by exploiting context-aware approaches for multimedia content adaptation. The dynamic model demonstrates the strong impact of the multimedia adaptation based on the user cognitive styles to available bandwidth condition.

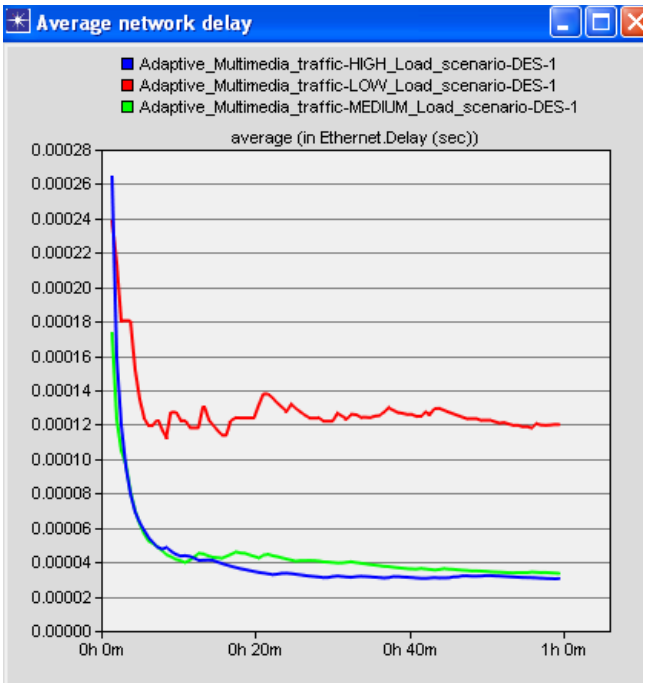


Figure 5. Average network delay analysis for three different network simulation scenarios

Results received from the simulation have confirmed the correctness of the newly proposed model for adaptive multimedia learning, and user experience is increased by using interactive learning process. In that way constant assessment is important to make appropriate adaptation of the learning content. Dynamic creation provides more personalized, context aware delivery and organization of the multimedia content types according to users learning style, knowledge progress and network conditions.

Challenging task is to analyze the influence of the context-aware conditions in the cloud computing environment and its influence in the process of personalized multimedia content generation.

REFERENCES

[1] P. Pankaj, V. Balkrishnan, and R. Camanathan, "SCORM for e-Learning: Towards implementation a collaborative learning platform." IEEE International Conference on Technology for Education. T4E 2010. Mumbai, India. pp. 236-238, 2010.

[2] H. Srimathi and S. K. Srivatsa, "SCORM-compliant personalized eLearning using Instructional Design principle." IEEE Computer Society. International Conference on Signal Processing Systems, Singapore, pp. 738-742, 2009.

[3] T.H. Wang, N. Y. Yen, Y.L. Du, and T. K. Shih, "Courseware Authoring Tool for Achieving Interoperability among Various E-Learning Specifications Based on Web 2.0 Technologies." ICPPW '07 Proceedings of the 2007 International Conference on Parallel Processing Workshops, Washington, DC, USA, IEEE Computer Society, doi:10.1109/ICPPW.2007.32. pp. 25, 2007.

[4] S. Y. Chen, G. Ghinea, and R. D. Macredie. "A cognitive approach to user perception of multimedia quality: An empirical investigation." International Journal of Human-Computer Studies. Volume 64 Issue 12, doi:10.1016/j.ijhcs.2006.08.010. pp. 1200–1213, December 2006.

[5] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. M. Sheppard, and Z. Yang. "Quality of Experience in Distributed Interactive Multimedia Environments: Toward a Theoretical Framework." MM '09 Proceedings of the 17th ACM international conference on Multimedia, Beijing, China, doi: 10.1145/1631272.1631338, pp. 481-490, 2009.

[6] ITU-T Report 2007. "Definition of Quality of Experience (QoE)", International Telecommunication Union, Liaison Statement, Ref.: TD 109rev2 (PLEN/12), January 2007.

[7] N . Stash and P.De Bra. "Incorporating cognitive styles in AHA! (The Adaptive Hypermedia Architecture)." Proceedings of the International Conference Web-Based Education (IASTED). Electronic Publication, pp. 378-383, 2004.

[8] E. Popescu, P. Trigano and C. Badica. "Adaptive educational hypermedia systems: A focus on learning styles." In Procedings of the International Conference on Computer as a Tool (EUROCON), Warsaw, Poland, IEEE Computer Society, 2007.

[9] SCORM- Shareable Content Object Reference Model. http://adlnet.org <accessed on 05/02/2012>

[10] OPNET Technologies, Inc. http://www.opnet.com/ <accessed on 08/02/2012>

[11] G. M. Muntean. "Efficient Delivery of Multimedia Streams Over Broadband Networks Using QOAS." IEEE Transactions on Broadcasting, Vol. 52, Issue. 2, pp. 230 – 235, June 2006.

[12] Y. Eltigani, A. Mustafa and S. M. Sharif. "An approach to Adaptive E-Learning Hypermedia System based on Learning Styles (AEHS-LS): Implementation and evaluation." International Journal of Library and Information Science Vol. 3, pp. 15-28, January 2011.

[13] K. R. Laghari, N. Crespi, B. Molina and C.E. Palau. "QoE aware Service Delivery in Distributed Environment." Workshops of International Conference on Advanced Information Networking and Applications, Biopolis, Singapore, IEEE Computer Society, doi:10.1109/WAINA.2011.58, pp. 837-842, 2011.

[14] R. E. Mayer. "The promise of multimedia learning: using the same instructional design methods across different media." Elsevier Science. Learning and Instruction, Volume: 13, Issue: 2, doi: 10.1016/S0959-4752(02)00016-6, pp. 125–139, 2003.

[15] Z. Lu and H. Yang. *Unlocking the Power of OPNET Modeler*. Cambridge University Press, January 2012.

[16] A. Scherp and S. Boll. "MM4U- A framework for creating personalized multimedia content." In: Srinivasan, Uma, Nepal, Surya (Hrsg.), Managing Multimedia Semantics, IRM Press, doi:10.1.1.59.6880, pp. 246-287, 2005.

 ISBN: 978-1-61208-219-6

Decentralized Probabilistic Auto-Scaling for Heterogeneous Systems

Bogdan Alexandru Caprarescu, Dana Petcu
Research Institute e-Austria and West University of Timişoara
Timişoara 300223, România
{*bcaprarescu, petcu*}*@info.uvt.ro*

***Abstract*—Scalability has become a de facto non-functional requirement for today's Internet applications that start small and aim to become a huge success. The ability of the system to automatically scale is required by the dynamic nature of the workload experienced by these applications. In this context, the DEPAS (Decentralized Probabilistic Auto-Scaling) algorithm assumes an overlay network of computing nodes where each node probabilistically decides to shut down, allocate one or more other nodes, or do nothing. DEPAS was formulated, tested, and theoretically analyzed for the simplified case of homogenous systems. In this paper, we extend DEPAS to heterogeneous systems. Thus, we provide a new formula for computing the node-level addition probability and evaluate it both theoretically and experimentally.**

***Keywords*-auto-scaling; decentralized computing; randomized algorithms; cloud computing**

I. INTRODUCTION

As the main characteristic of cloud computing, the on-demand provisioning of hardware resources creates the premises for a theoretically infinite scalability of services deployed in the cloud [1]. In this context, providing a software system with the capacity to automatically scale in order to accommodate the fluctuations of the workload has become an interesting topic in both academia and industry. On one hand, the researchers focus on complex solutions that optimize the resource consumption and the QoS of the services. On the other hand, some cloud providers, such as Amazon EC2 and Rightscale, offer policy-based auto-scaling solutions that can be easily configured by their customers.

A common characteristic of the large majority of research and industrial solutions for auto-scaling consists in their centralization. In this way, the infinite scalability that is theoretically possible is jeopardized by the central component running the auto-scaling algorithm. This is not only a theoretical problem and the limitations of centralized management were experienced with industrial systems such as VMware [2]. Moreover, a central manager acts as a single point of failure, too.

Our aim is to provide an auto-scaling algorithm that is both scalable and fault tolerant. By being fault tolerant we mean that the algorithm should continue to work in the presence of node churn and message losses in the inter-node communication. To achieve this goal, we take inspiration from the unstructured P2P systems, which proved to be highly scalable and robust [3]. Thus, in [4], we described DEPAS, a decentralized probabilistic auto-scaling algorithm. DEPAS assumes an overlay network of nodes. Each node is a virtual machine that runs the same software comprising the functional service and a few non-functional components: overlay manager, load balancer, and auto-scaler. Each non-functional component runs a decentralized algorithm. The auto-scaler runs DEPAS.

The parameters of DEPAS are the desired load, L_0, and the load variation, δ. The goal of DEPAS is to maintain the average load of the system in the interval $(L_0 - \delta, L_0 + \delta)$. To do that, each node estimates the average load of the system and, if it is not within that interval, then the node probabilistically executes the appropriate scaling action (i.e., remove itself if the load is lower than $L_0 - \delta$ or allocate additional nodes if the load is higher than $L_0 + \delta$). The main problem is how to compute the node-level scaling probability.

DEPAS was originally formulated and tested for the simplified case of homogenous systems, in which all nodes have the same capacity [4]. The link between δ, number of nodes, and the probability of allocating the right number of nodes was theoretically analyzed and we provided algorithms for finding either the minimum δ (for a given number of nodes) or the minimum number of nodes (for a given δ) so that a minimum correctness probability is guaranteed [5].

Some tests with heterogeneous nodes were also performed in [4] under the assumption that each node randomly choses the capacity of the node to be added with the same distribution as the capacity distribution of the existing nodes. For example, in a system with 70% nodes with a low capacity and 30% nodes with a high capacity, a newly allocated node would have 70% chances to be a low capacity one and 30% chances to be a high capacity node. However, this is a highly constrained scenario for which it is difficult to imagine a practical applicability. This is because in practice the type of a new virtual machine is chosen so that to optimize a certain client-defined criterion (e.g., the VM type with the minimum cost per capacity unit). The selection of the optimal node type may be related to the criterion to select it, certain aspects of the cloud infrastructure, or the bid for resources from other customers, but it has nothing to do with the capacity distribution of the existing nodes.

Therefore, in this paper, we extend DEPAS to heterogeneous systems without imposing any constraint on the

 ISBN: 978-1-61208-219-6

capacity distribution of the new nodes. Thus, we provide a formula for computing the addition probability that works no matter the capacity of the new nodes. The correctness of the formulas for computing the addition and removal probabilities is proven both theoretically and experimentally.

The remaining of this paper is organized as follows. The DEPAS algorithm for heterogeneous systems is described in Section II and experimentally verified in Section III. Related work is discussed in Section IV, while Section V concludes the paper.

II. DEPAS FOR HETEROGENEOUS SYSTEMS

We assume a system composed of n nodes with capacities $C_i, i = 1..n$. The capacity of a node is the maximum number of requests per second that can be processed by the service deployed on that node. Let C be the total capacity of the system (the sum of the capacities of all nodes). A complete list of notations is given in Table I. The load of a node, noted with L_i, is computed at a given moment in time as a ratio between the average number of requests per second that were either processed or rejected by that node over a certain timeframe and the capacity of the node. Depending on the load balancing algorithm, a node may reject a request in certain cases (e.g., when the system is overloaded and the maximum response time of the request can not be met). Then, the average load of the system, L, is computed as a ratio between the workload of the system and the capacity of the system as expressed by equation (1). Note that in the case when the workload received by the system overcomes its capacity, the average load is supra-unitary.

$$L = \frac{\sum_{i=1}^{n} L_i C_i}{\sum_{i=1}^{n} C_i} = \frac{\sum_{i=1}^{n} L_i C_i}{C} \quad (1)$$

The DEPAS algorithm for heterogeneous systems is shown in Algorithm 1. It is periodically run with period T by each node and begins by retrieving an estimation of the average load of the system. Note that the average load is not computed at this time, but just retrieved from the component running the average protocol. If the load is less than or equal to $L_0 - \delta$, then the node computes a removal probability indicator using formula (2) and, because the indicator is sub-unitary in this case, the node uses it as the probability to remove itself. Otherwise, if the load is higher than or equal to $L_0 + \delta$, then the node obtains its own capacity and the capacity of the node type that is the most convenient to be allocated at this time. Then, the probability indicator is computed using formula (3). In this situation, the indicator can be supra-unitary, where its integer part represents the number of nodes to be added for sure, while its fractional part is used as the probability to add another node. Note that the *random()* function generates a uniformly distributed random decimal number between 0 and 1.

$$pi_i^{rem} = \frac{L_0 - L_i^\star}{L_0} \quad (2)$$

Algorithm 1 DEPAS for Heterogeneous Systems

```
while true do
  wait(T)
  L_i* ← getEstimatedAverageSystemLoad()
  if L_i* ≤ L_0 − δ then
    pi_i^rem ← computeRemovalProbInd(L_i*, L_0)
    p_i ← pi_i^rem
    if p_i < random() then
      removeSelf()
    end if
  else
    if L_i* ≥ L_0 + δ then
      C_i = getSelfCapacity()
      C_i^add = computeNewNodesCapacity()
      pi_i^add ← computeAdditionProbInd(L_i*, L_0, C_i, C_i^add)
      m ← ⌊pi_i^add⌋
      p_i ← {pi_i^add}
      if p_i < random() then
        m ← m + 1
      end if
      addNodes(m, C_i^add)
    end if
  end if
end while
```

$$pi_i^{add} = \frac{L_i^\star - L_0}{L_0} \frac{C_i}{C_i^{add}} \quad (3)$$

The remaining of this section proves that the formulas (2) and (3) are correct. They are correct if the expected capacity to be removed or added is equal to the optimal capacity to be removed or added, respectively. The optimal capacity is

Table I
DEPAS NOTATIONS

T	The duration of a DEPAS cycle (in seconds)
n	Number of nodes of the system
C	Total capacity of the system
C_i	Capacity of node i
C_i^{add}	Capacity of the nodes to be added by node i in the current cycle
C_{opt}^{rem}	Optimal capacity to be removed
C_{opt}^{add}	Optimal capacity to be added
C_{exp}^{rem}	Expected capacity to be removed
C_{exp}^{add}	Expected capacity to be added
L_0	Desired load threshold (percent with respect to the capacity)
δ	Defines the allowed load variation
L_i	Load of node i (percent with respect to node capacity)
L	Average load of the system (percent with respect to the capacity)
$L_i^\star$	An estimation of the average load of the system done by node i
pi_i^{add}	Addition probability indicator computed by node i
pi_i^{rem}	Removal probability indicator computed by node i
p_i	Node-level probability computed by node i

defined as the amount of capacity that needs to be subtracted from or added to the system so that the new average load is equal to the desired load. Theorems 1 and 2 provides formulas for computing the optimal capacity to be removed and the optimal capacity to be added, respectively.

Theorem 1. *Let $L_0 \in (0,1)$ be the desired load. Consider a system with total capacity C and average load $L < L_0$. Then, the optimal capacity to be removed from the system is computed as follows:*

$$C_{opt}^{rem} = \frac{L_0 - L}{L_0} C$$

Proof: As the system has the same workload before and after removing capacity we have $LC = L_0(C - C_{opt}^{rem})$, from where it results the formula given by the theorem. ■

Theorem 2. *Let $L_0 \in (0,1)$ be the desired load. Consider a system with total capacity C and average load $L > L_0$. Then, the optimal capacity to be added to the system is computed as follows:*

$$C_{opt}^{add} = \frac{L - L_0}{L_0} C$$

Proof: Analogues with the proof of Theorem 1. ■

The expected capacity to be removed/added is computed for a cycle of DEPAS. A cycle has a duration of T seconds, in which each node runs DEPAS exactly once. In order to be able to compute the expected capacity we assume that each node precisely estimates the average load of the system, which means that $L_i^\star = L \forall i = 1..n$. Under these considerations, theorems 3 and 4 prove the correctness of the formulas for computing the removal and addition probabilities.

Theorem 3. *Let $L_0 \in (0,1)$ be the desired load. Consider a system with n nodes, total capacity C and average load $L < L_0$. The expected capacity to be removed in a cycle of DEPAS is equal to the optimal capacity to be removed.*

Proof:

$$C_{exp}^{rem} = \sum_{i=1}^{n} \frac{L_0 - L}{L_0} C_i = \frac{L_0 - L}{L_0} C = C_{opt}^{rem}$$

■

Theorem 4. *Let $L_0 \in (0,1)$ be the desired load. Consider a system with n nodes, total capacity C and average load $L > L_0$. The expected capacity to be added in a cycle of DEPAS is equal to the optimal capacity to be added.*

Proof:

$$C_{exp}^{add} = \sum_{i=1}^{n} \frac{L - L_0}{L_0} \frac{C_i}{C^{add} i} C^{add} i = \frac{L - L_0}{L_0} C = C_{opt}^{add}$$

■

In a previous paper [4], the addition probability indicator was computed with formula (4). We are interested in which conditions this formula leads to a correct allocation in a heterogeneous systems. A correct allocation happens when $C_{exp}^{add} = C_{opt}^{add}$. By replacing the expected and optimum capacities with their formulas and making the simplifications it results equation (5).

$$pi_i^{add} = \frac{L - L_0}{L_0} \tag{4}$$

$$\sum_{i=1}^{n} C_i^{add} = C \tag{5}$$

From equation (5) it turns out that formula (4) can be used to compute the addition probability in a heterogeneous system only if the sum of the potential capacities to be added by each node is equal to the total capacity of the system. This is obviously a very particular situation without any practical motivation.

Therefore, in this section, we provided a general formula – expressed by equation (3) – for computing the addition probability of the DEPAS algorithm in a heterogeneous system.

III. Experimental Results

In the previous section, we proved that the formulas for computing the removal and addition probabilities are correct providing that each node knows the average load of the system. In this section, we relax this requirement and experimentally show that those formulas lead to good allocations even if the average load of the system is approximated at each node with the average load of the node and its neighbors.

For running the experiments, we adapted the simulator that was used in a previous paper [4]. This simulator is based on the Protopeer library [6] and is available for download [7]. The simulator is described in Subsection III-A, while the results of the experiment are shown in Subsection III-B.

A. Settings

In the simulator, we designed three types of peers: client, entry point, and worker. In the experiment that is described in this section we use only one client and one entry point. The client issues requests according to an exponential distribution whose mean value follows a given workload track (see Subsection III-B). The requests arrive at the entry point, which knows a percent of all workers but not less than a given minimum. The workers known by the entry point are randomly selected and periodically renewed. The entry pointed dispatches each request to one worker according to a capacity-weighted random load balancing strategy. The values used in the experiment for the parameters that are discussed in this subsection are listed in Table II.

 ISBN: 978-1-61208-219-6

Table II
PARAMETERS OF THE SIMULATOR

Min no of entry point neighbors	50
Percent of entry point neighbors	2%
Entry point neighbor reshuffle period	120s
Overlay degree	50
Overlay management cycle	0.5s
Max queue size	3
Max no of hops	10
Mean execution time	1
Load monitoring period	60s
T (DEPAS cycle duration)	60s
L_0	0.7
δ	0.1

The nodes are organized into an unstructured overlay network where each node runs an overlay management algorithm to maintain a list of neighbors. We adapted the gossip-based overlay management algorithm developed by Jelasity et al. [8] to obtain two characteristics needed by our system: quick removal of links to dead nodes (needed by both load balancing and average load estimator) and low-deviation in-degree (needed for proper load balancing). The resulted overlay management algorithm will be described in a dedicated paper. The values of the main parameters related to overlay management (overlay degree and cycle duration) are given in Table II.

The first-level load balancing performed by the entry point is accompanied by a decentralized load balancing that is performed by the workers. The decentralized load balancing algorithm is taken from [9] and adapted for heterogeneous systems. The idea is that, when a worker receives a request (which may come from either an entry point or other worker), an admission function is called to decide whether the request is scheduled on the current node, routed to a neighbor, or rejected. More concretely, each node uses an internal queue to store the requests that are pending for execution and prioritize them depending on the time they were issued by the client (i.e., older requests get priority over newer requests). If the length of the queue divided by the capacity of the node is higher than a threshold then the request is added to the queue. Otherwise, if the number of nodes already visited by the request without being scheduled on neither of them (called number of hops) is less than a threshold, then the request is routed to a neighbor of the current node in the overlay network. The neighbor is selected using the same capacity-weighted random strategy that is used by the entry point. Finally, if the maximum number of hops is reached, then the request is rejected.

The request execution time is exponentially distributed with a constant mean. The capacity of a node is expressed in the simulator using an integer number c, which means that the node can execute c requests in parallel. The mean execution time is always the same no matter the capacity of the node.

The average load of a node is computed per second by counting the requests that were either processed or rejected by the current node over a timeframe called load monitoring period. The average load of the system is approximated with the average load of the current node and its neighbors. The load monitoring period and the DEPAS cycle duration (i.e., the time between two consecutive executions of DEPAS on the same node) are important parameters because they affect the reactivity and accuracy of DEPAS. On one hand, small values of these parameters make DEPAS to quickly react to workload changes but also very sensible to oscillations (i.e., additions and removals of nodes mixed in a row). Of course, the oscillations should be avoided because they waste the money of the customer. On the other hand, higher values of the load monitoring period and the DEPAS cycle make the system more stable at the cost of delaying its reactivity and thus rejecting more requests when confronted with workload bursts.

Finally, the values of the desired load and load variation threshold are also given in Table II.

B. Results

Provided that a list of fresh neighbor information is available at each node and that the system is properly load balanced, the DEPAS algorithm works properly and is very scalable and fault tolerant. It is very scalable because (i) it is very simple and is run once every T seconds and (ii) it does not require any message exchange between neighbors. It is robust because it does not care about node crashes as the update of neighbors information is the task of the overlay management algorithm. Therefore, a complete solution based on DEPAS is scalable and robust as long as it employs scalable and robust overlay management and load balancing algorithms. But the scalability and robustness of these algorithms were already experimentally proved in [4]. This is why, in this paper, we experimentally check only the correctness of the DEPAS algorithm for heterogeneous systems. More concretely, for a dynamic workload scenario we verify that the allocated capacity is close to the optimum capacity.

The overlay management, load balancing, and auto-scaling algorithms are non-deterministic. Therefore, meaningful results can be obtained only by averaging the results of several identical, but independently performed experiments. The results presented in this paper are the average results of 32 identical experiments. All experiments were sequentially executed on one Amazon High-CPU Medium Instance (1.7 GB of memory, 2 virtual cores with 2.5 EC2 Compute Units each) running Amazon Linux 64-bit.

Figure 1 shows the mean workload track used in our experiment. The actual workload is exponentially distributed with the dynamic mean given by the workload track. The simulated experiment lasts 2600 seconds. The mean workload is 70 requests/s at the beginning of the experiment and

 ISBN: 978-1-61208-219-6

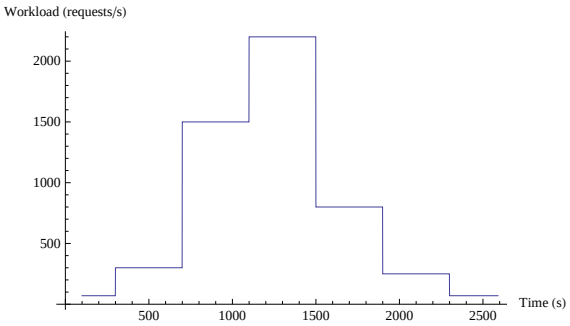


Figure 1. Mean workload track

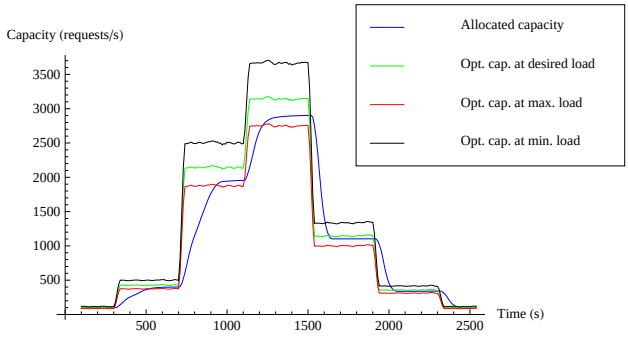


Figure 2. Allocated vs. optimal capacity

increases in a few steps to reach 2200 requests/s at its peak. Then, to also test the scale out branch of DEPAS, the mean workload is decreased in steps.

We use two types of nodes: low capacity nodes with a capacity of 1 request/s and high capacity nodes with a capacity of 5 requests/s. The system is initialized with 100 low capacity nodes, which have the perfect capacity for handling the initial workload of 70 requests/s. The existing nodes can add high-capacity nodes in the first 1099 seconds and low capacity nodes after second 1100 inclusive (which marks the last workload burst as shown in Figure 1).

Figure 2 shows the capacity allocated by DEPAS versus the optimum capacity computed for the desired load (L_0), min load ($L_0-\delta$), and max load ($L_0+\delta$). It can be seen that, after a period of adaptation, DEPAS allocates a capacity that is between the optimum capacity at max load and optimum capacity and min load. The delay in adaptation is caused by the duration of the DEPAS cycle (60 seconds) and by the fact that the load is averaged over the last 60 seconds. But, as explained in Subsection III-A, the good side of these setting consists in the fact that, despite its randomized and decentralized nature, DEPAS is very stable and did not cause any capacity oscillation.

Figure 3 shows the variation of the total number of nodes, number of low capacity nodes, and number of high capacity nodes. We can see that the system, initially composed of 100 low capacity nodes, adapts to the increasing workload

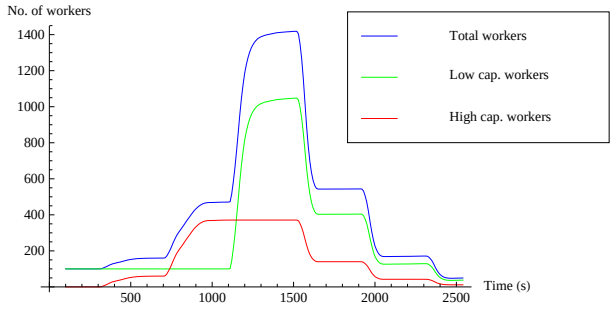


Figure 3. Number of nodes

by adding high capacity nodes before the second 1100 and low capacity nodes afterwards. Then, during the scale out, the system removes both low capacity and high capacity nodes. As expected, the percent of both types of nodes out of the total number of nodes seems to remain constant while scaling out. Figure 3 confirms that DEPAS does not make any over-provisioning (i.e., allocation of more nodes that needed) or over-de-provisioning (i.e., removal of more nodes than needed).

In conclusion, the experimental results show that, facing a variable workload in a heterogeneous system, DEPAS allocates the right capacity even if each node works with a local approximation of the average load.

IV. Related Work

In this paper, we discussed a few other decentralized approaches to auto-scaling. A detailed state of the art on autonomic resource provisioning for cloud computing can be found in [4].

A decentralized economic-inspired solution to the auto-scaling of component-based systems was proposed by Bonvin et al. [10]. They use a multi-agent approach, in which each server is managed by a server agent. This agent makes decisions related to the migration/replication/removal of the components deployed on that server. The problem is that each agent stores a complete mapping (maintained through gossiping) of components and servers. In other words, each agent has a complete view of the system. Because of that, although the approach is decentralized in the sense that there is no central manager, it is not scalable with respect to the number of components and servers.

Another decentralized auto-scaling approach was proposed by Wuhib et al. [11]. They aim to develop a Platform as a Service for hosting sites in the cloud. In their approach, each virtual machine is managed by a VM manager, which is connected through a custom overlay network to other VM managers that store instances of the same sites. The utility of a site instance is computed as the ratio between the allocated CPU capacity and the CPU demand. The utility of the system is the minimum utility of all instances of all sites.

 ISBN: 978-1-61208-219-6

A decentralized heuristic algorithm is used to maximize the utility of the system while minimizing the cost of adaptation. The resulting system is scalable with respect to the number of virtual machines and the number of sites, but it is not scalable with respect to the number of instances of a site.

Montresor and Zandonati proposed a decentralized algorithm for selecting a slice of a P2P system [12]. This slice may contain nodes with given characteristics that are needed for running a certain distributed application. Their approach shares the same idea with DEPAS: each node probabilistically decides to join the slice or depart from the slice. But their approach is more complex and less scalable than DEPAS because they use an epidemic broadcast algorithm to inform all nodes about the slice to be created and a peer counting algorithm that provides each node with an estimation of the slice size. The key to the high scalability of DEPAS is that a node does not need to know either the total number of nodes or the total capacity of the system.

V. Conclusion and Future Work

The Decentralized Probabilistic Auto-Scaling (DEPAS) algorithm can be used to deploy large-scale service systems whose scalability is limited only by the amount of virtualized resources that can be rented from IaaS providers. DEPAS assumes that the computing nodes are organized into an unstructured overlay network and run a scalable load balancing algorithm. DEPAS is run by each node, which probabilistically decides to scale in and out. The main problem in DEPAS is how to compute this probability.

In [4] and [5], DEPAS was formulated, tested, and theoretically analyzed for the simplified case of homogenous systems. In this paper, we provided scaling probabilities formulas that work in the general case of heterogeneous systems. We proved both theoretically and experimentally that, by using the proposed formulas, DEPAS reacts to workload variations by allocating the right capacity in the first place.

DEPAS implementation will be included in the near future in the elasticity mechanism of the second version of the open-source and deployable Platform-as-a-Service named mOSAIC (www.mosaic-cloud.eu, bitbucket.org/mosaic) and will be the starting point in the research activities related to auto-scaling that are foreseen in MODAClouds (www.modaclouds.eu).

Acknowledgment

This research has been partially funded by the Romanian National Authority for Scientific Research, CNCS UEFISCDI, under project PN-II-ID-PCE-2011-3-0260 (AMICAS) and by the European Commission, under project FP7-ICT-2009-5-256910 (mOSAIC). Bogdan Caprarescu is partially supported by IBM through a PhD Fellowship Award.

We would like to thank to Nicola Calcavecchia and Daniel Dubois for their contribution to the development of the DEPAS simulator.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50–58, April 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[2] S. Meng, L. Liu, and V. Soundararajan, "Tide: achieving self-scaling in virtualized datacenter management middleware," in *Proceedings of the 11th International Middleware Conference Industrial track*, ser. Middleware Industrial Track '10. New York, NY, USA: ACM, 2010, pp. 17–22. [Online]. Available: http://doi.acm.org/10.1145/1891719.1891722

[3] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1-4, pp. 72–93, 2005.

[4] N. M. Calcavecchia, B. A. Caprarescu, E. Di Nitto, D. J. Dubois, and D. Petcu, "Depas: A decentralized probabilistic algorithm for auto-scaling," *CoRR*, vol. arXiv:1202.2509, 2012.

[5] B. A. Caprarescu, E. Kaslik, and D. Petcu, "Theoretical analysis and tuning of decentralized probabilistic auto-scaling," *CoRR*, vol. arXiv:1202.2981, 2012.

[6] W. Galuba, K. Aberer, Z. Despotovic, and W. Kellerer, "Protopeer: a p2p toolkit bridging the gap between simulation and live deployement," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09, ICST, Brussels, Belgium, Belgium, 2009, pp. 60:1–60:9.

[7] "Simulator of depas for heterogenous systems," 2012, http://bogdan.softinvent.org/research/depas/ (accessed Mar 11 2012).

[8] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Trans. Comput. Syst.*, vol. 23, pp. 219–252, August 2005. [Online]. Available: http://doi.acm.org/10.1145/1082469.1082470

[9] C. Adam and R. Stadler, "A middleware design for large-scale clusters offering multiple services," *IEEE Transactions on Network and Service Management*, vol. 3, no. 1, pp. 1–12, 2006.

[10] N. Bonvin, T. G. Papaioannou, and K. Aberer, "Autonomic sla-driven provisioning for cloud applications," in *Proceedings of the 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2011, pp. 434–443.

[11] F. Wuhib, R. Stadler, and M. Spreitzer, "Gossip-based resource management for cloud environments," in *Proceedings of the 2010 International Conference on Network and Service Management (CNSM)*, 2010, pp. 1–8.

[12] A. Montresor and R. Zandonati, "Absolute slicing in peer-to-peer systems," in *Proc. of the 5th Int. Workshop on Hot Topics in Peer-to-Peer Systems (HotP2P'08)*. Miami, FL, USA: IEEE, Apr. 2008.

 ISBN: 978-1-61208-219-6

Dynamic Adaptation of Opportunistic Sensor Configurations for Continuous and Accurate Activity Recognition

Marc Kurz, Gerold Hölzl, Alois Ferscha
Johannes Kepler University Linz
Institute for Pervasive Computing
Linz, Austria
{*kurz, hoelzl, ferscha*}*@pervasive.jku.at*

***Abstract*—An ever-larger availability of devices that are attached with different sensing capabilities (e.g., smart phones) shifted the challenge in activity and context recognition from the application specific deployment of new sensors to the utilization of already available devices. Therefore, a system that operates in an opportunistic way has to take advantage of the currently available sensing infrastructure in terms of utilizing sensors in form of ensembles that are best suited to execute a specific activity recognition task. Continuous, stable, and accurate activity recognition can be assured if such a system is able to react in real-time to such dynamics in the sensing infrastructure. In detail, this paper tackles the characteristic application cases where sensors spontaneously appear, disappear and reappear in the sensing infrastructure and evaluates the continuousness and stability of the self-adaption methods within the OPPORTUNITY Framework, which is a reference implementation of an opportunistic activity and context recognition system.**

Keywords-Opportunistic sensing; activity and context recognition; self-adaptation.

I. INTRODUCTION

Opportunistic activity recognition is characterized by the fact that sensor systems that gather environmental data to infer people's activities are not presumably known at design time of the system [1]. Actually, the activity and context recognition system that operates in an opportunistic way has to take advantage of the currently available devices in order to execute the recognition task as accurate as possible. Another crucial characteristic is that the recognition purpose (i.e., *recognition goal*) is not fixed at design time of the system, either. This goal can be rather defined by a user or an application at runtime of the system [1]. Subsequently, the currently available sensor devices (i.e., the *sensing infrastructure*) have to be queried, and the set of sensors has to be identified that is able to contribute to the recognition goal. According to such a recognition goal, the system configures *ensembles*, which are (sub-)sets of available sensors that are best suited to execute this specific recognition goal [2]. Since the sensor systems that are involved in such ensembles are not presumably known, the system has to dynamically handle changing sensor environments, which also includes different modalities and types of sensors [3]. These characteristics of an opportunistic activity and context recognition systems allow the identification of application cases (see also [4]): (i) *sensor appears*, (ii) *sensor disappears*, (iii) *sensor reappears*, (iv) *sensor delivers reduced-quality data*, and (v) *sensor learns from other sensors*.

This paper tackles the application cases (i), (ii), and (iii), where sensors spontaneously appear, disappear and reappear in the surrounding sensing infrastructure. The remaining two application cases (iv) and (v), where on the one hand the reduced quality data and on the other hand the enhancement of sensor capabilities at runtime are considered have already been subject of discussion and evaluation in recent publications (see [4] and [5]). The experiments and evaluations that are contained in this paper try to answer the question whether the developed concepts like *sensor self-descriptions* (see [1][4][5]), that describe the sensor from a meta-level with respect to the recognition capabilities for specific goals and that enable the dynamic instantiation of activity recognition chains, and the self-organization concepts that enable the dynamic configuration of senor ensembles [5] ensure a continuous, stable and highly accurate activity recognition. Therefore, the paper utilizes a rich dataset that was recently recorded in a kitchen scenario [6]. This allows a high quality evaluation in a repeatable simulated setting with a publicly available dataset. Furthermore, the same setup is also evaluated in an experiment with physical sensor systems to demonstrate the real-time capabilities of the concepts and the OPPORTUNITY Framework, which is a reference implementation of an opportunistic activity and context recognition system together with the sensor self-description and the ensemble configuration concepts.

The rest of the paper is structured as follows. Section II provides an overview of related work. Section III provides a detailed description of the opportunistic activity recognition approach, whereas the main focus lies on the application cases that build the core for further evaluations. Section IV presents an experimental setup based on a rich dataset and evaluates the stability and steadiness of the OPPORTUNITY Framework in terms of dynamically configuring sensor ensembles according to a recognition goal. The last Section V closes with a conclusion and an outlook.

 ISBN: 978-1-61208-219-6

II. Related Work

Activity and context recognition is a research topic that has been tackled from different groups in the last years and decade(s). Mantyjarvi et al. [7] and Bao and Intille [8], present the principles of activity recognition with acceleration sensors mounted on different parts of the body of subjects. The recognition of human activities with a defined set of sensors, with fixed (on-body) position and location is evaluated and tested with different algorithms (see also [9][10][11] as recent examples). The novelty of this paper is the fact that the sensing infrastructure is not presumably known by the system and thus has to react on spontaneous changes to the availability of the surrounding sensors.

Concerning the characteristic application case of such an opportunistic system where sensors that are involved in the recognition process deliver faulty or quality-reduced data (e.g., due to a low battery level), this anomaly can be detected, as described in [1]. This approach is demonstrated in the OPPORTUNITY Framework, which is a working reference implementation of an opportunistic activity and context recognition system, in [5]. There, a quantitative measure called *Trust Indicator* was used to weight the reduced data quality. The second application case that was already demonstrated within the OPPORTUNITY Framework is the crucial task of autonomously enhancing a single sensor's capabilities by observing an active ensemble. This method (also referred to as *transfer learning*) is described in detail in [12] and demonstrated in a running system and tested with respect to real-time requirements in [4]. The remaining characteristic cases that deal with the spontaneous availability of sensors are the core subject of this paper.

Related work that concerns about activity recognition with spontaneously changing sensing environments is - to our best knowledge - very scarce. Approaches exist where sensors are dynamically selected in order to find an accuracy-power trade off [13], or where the activity recognition chain is ported to a sensor platform and distributed to multiple nodes in a wireless sensor network [14].

Chavarriaga et al. [15] present an approach that is derived from an information theoretic concept, where available sensors are dynamically picked with respect to the expected recognition performance of the sensor aggregation. This approach works with the diversity measurements of possible classifier combinations and suffers from the known problem that a ground-truth in the first place is inevitable. Villalonga et al. [16] present a similar use case, where sensor ensembles are configured and the expected recognition accuracy is predicted. These publications are related in a way that not all sensors that would be available are integrated in a sensor ensemble for a specific recognition task but a subset of them, which is optimized in terms of performance. Nevertheless, this paper tackles the problem of handling spontaneous changes in configured sensor ensembles.

III. Opportunistic Activity Recognition

Opportunistic activity and context recognition arises as new working principle since sensor devices have recently become more and more integrated into the daily life. This shifted the effort from the application specific deployment of sensors for specific recognition tasks to the utilization of already available sensors. Therefore, in preceding publications, we have already introduced the concept of sensor abstractions (see [1][3]), which enables the common usage of material as well as immaterial devices as general type *sensor*. This is an important feature in an opportunistic system, since the sensor type and modality cannot be predefined and thus has to be able to handle open-ended sensor environments. The second important method that has already been introduced in recent papers (see [1][4][5]) is the concept of sensor *self-descriptions*. The standardized XML documents are key components in an opportunistic system and provide a two dimensional description of a sensor. The description consists of (i) a technical part that describes the physical working characteristics from the sensing device (could be seen as transcript from the technical specification), and of (ii) a dynamic part that encapsulates the sensor capabilities in terms of recognizing activities. Both parts of the sensor self-description are composed of *SensorML* [17], which is an approved standard. Besides inevitable parts like identifiers, information about the sensor's position, and other relevant information, the dynamic part of the sensor self-description contains key elements that enable the handling of dynamically changing sensor ensembles: *ExperienceItem*s (see also [1][4][5]). The *ExperienceItem*s contain all required building blocks for a dynamic configuration of an activity recognition chain for the dedicated sensor (i.e., (i) *FeatureExtraction*, (ii) *Classifier*, (iii) the accompanying *Classifier Model*, and (iv) the expected/estimated accuracy in form of the *Degree of Fulfillment*). *ExperienceItem*s are highly dynamic elements and can even be added or modified by the OPPORTUNITY Framework at runtime of the system. The application case *"Sensor Learns"*, where a sensor learns the recognition of a specific activity at runtime was demonstrated in [4]. The sensor is able to preserve experience (thus the name *ExperienceItem*), which can be gathered by training the required machine-learning technologies at system's runtime by comparing the sensor readings with the label output of available sensors that are configured in ensembles. In [5], the application case *"Sensor delivers faulty Data"* was demonstrated, which also relies on the capabilities of *ExperienceItem*s to be dynamically updated at runtime. This paper discusses and evaluates the remaining application cases, *"Sensor appears"*, *"Sensor disappears"*, and *"Sensor reappears"*.

Especially the dynamic description with the *ExperienceItem*s enables the dynamic configuration of ensembles according to a recognition goal and permits the required

 ISBN: 978-1-61208-219-6

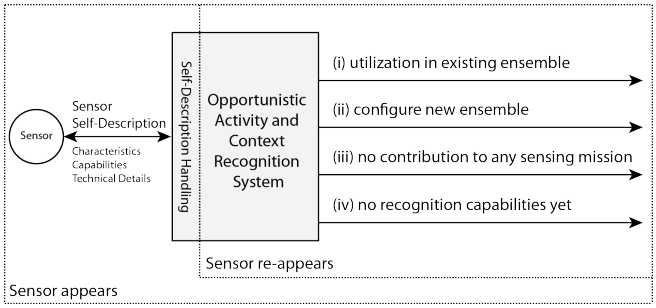


Figure 1. Illustration of the system reaction to the two application cases (i) *sensor appears* and (ii) *sensor disappears*.

stability, continuity and adaptability of the opportunistic activity recognition. Whenever a new sensor appears in the sensing environment, its accompanying (dynamic) self-description is queried and read to be aware about the sensor's present capabilities and its utilization in currently active ensembles. Upon the capabilities according to the dynamic description, the system then decides on the further utilization of the sensor, whereas four different possibilities can be distinguished, as summarized in Figure 1: (i) the sensor is integrated in a running ensemble, (ii) a new ensemble is configured containing the newly appeared sensor, (iii) the sensor's current capabilities do not match an active ensemble, thus the sensor cannot be used, and (iv) the sensor yet does not have any recognition capabilities (i.e., *ExperienceItem*s). Whenever the sensor is not integrated in an ensemble, because the capabilities are not sufficient, or not existing, the sensor could be a candidate to enhance its capabilities by applying *transfer learning* [4][12]. In the case the sensor was already online in the sensing infrastructure, thus is already known by the system on re-appearance, the querying and parsing of the self-description is obsolete. Nevertheless, the four consequent options of further utilization are equal to the aforementioned case (see also Figure 1).

When a sensor disappears - which is the third application case of interest for this paper - three different subsequent system steps have to be distinguished: (i) the sensor was not involved in an active ensemble, therefore no further action is necessary, (ii) the sensor was the exclusive component of an ensemble, thus the execution of the recognition goal cannot be continued, and (iii) the sensor was part of a bigger ensemble, there the execution can be continued with a (probably) reduced recognition rate (dependent on the disappearing sensor's performance). These application cases, which tackle the spontaneous availability of sensors in an opportunistic activity and context recognition system ((i) *sensor appears*, (ii) *sensor disappears*, and (iii) *sensor reappears*) are tested and evaluated in the OPPORTUNITY Framework in an experimental setting that relies on a pre-recorded dataset [6] where physical, on-body mounted sensor devices (see next Section IV) are used.

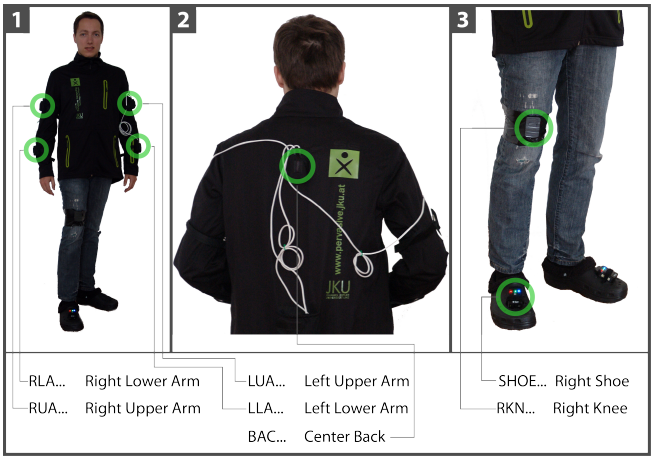


Figure 2. The on-body sensors for the experiment. The setting is similar to [4].

IV. Experiment Setup and Evaluation

The OPPORTUNITY Framework [1] is a reference implementation that realizes the aforementioned concepts of sensor abstractions and sensor self-descriptions. Furthermore, it is capable of executing activity recognition in real-time by applying different sensors, or to process repeatable simulation runs with pre-recorded sensor data. Therefore, the OPPORTUNITY dataset was recorded [6], which combines 72 sensors with 10 different modalities mounted on the body of persons, on objects (e.g., cup, knife, etc.) and in the environment (e.g., fridge, drawer, etc.) in a kitchen scenario. Each of these sensors can be replayed in the OPPORTUNITY Framework as *PlaybackSensor* [3], therefore act as it would be actually available. This allows the generation of repeatable simulation scenarios for evaluation and testing purposes. The experimental setting for this paper is meant to demonstrate the three application cases that tackle the sensors' spontaneous availability (*(re-)appear* and *disappear*). The chosen sensors are all mounted on the body of the test person. The upper body is equipped with 5 sensors of type *Xsens MTx*, which provides 3D acceleration. The sensors are located on both arms (right/left lower/upper arm - *RLA, RUA, LLA, LUA*), and on the upper back (*BAC*). The lower body is equipped with a self-composed bluetooth acceleration sensor on the right knee (*RKN*) and a *SunSPOT* (Small Programmable Object Technology) sensor on the right instep of the foot (*SHOE*). All sensors were operating with a sampling frequency of 100Hz and delivered 3D acceleration data (x-, y-, z-axes). The on-body sensor placement is illustrated in Figure 2.

The challenge is to demonstrate and evaluate the continuity and steadiness of the opportunistic activity and context recognition system by dynamically adapting to different conditions in the sensing infrastructure due to the sensors' spontaneous availability. This is done by dynamic configu-

 ISBN: 978-1-61208-219-6

Table I
OVERVIEW OF ACCURACIES OF THE SINGLE SENSORS FOR THE RECOGNITION OF THE MODES OF LOCOMOTION.

Sensor	RKN	SHOE	LUA	LLA	RUA	RLA	BAC
Accuracy	0.604	0.698	0.858	0.719	0.769	0.709	0.761

ration of different sensor aggregations in form of ensembles over a certain amount of time. We decided to use a rather simple set of activities since not the handling of complex tasks is the major contribution but the accurate and stable activity recognition even if the sensing environment changes. The activities of interest in the exemplary scenario are fixed to the modes of locomotion (i.e., *WALK*, *STAND*, *SIT*, *LIE*). Each of the seven involved on-body sensors is trained to recognize these four activities initially by utilizing the available ground-truth in the dataset (as already mentioned, the training could have also been done by applying the transfer learning [4] approach, but to ensure that all seven sensors have the same conditions, the initial training phase was selected). As features the *mean* and *variance* were used, the classification method was set to be the *NCC* classifier (reasons for these choices are (i) the ease of computation and (ii) the potentially good recognition results [18]). The resulting accuracies for the single sensors were calculated by comparing the predicted activity classes to the actual activity classes after the training phase, and are listed in Table I.

Each sensor's self-description was enhanced with an *ExperienceItem* that contains every piece of information (including the feature extraction method, the classifier method together with the required and pre-trained classifier model in form of a JSON file (see [5] for an example), and the sensor position) to dynamically configure the activity recognition chain at system's runtime. The combination of multiple sensors (respectively multiple recognition chains) in the experiment session is done by applying *MajorityVoting* fusion. This rather simple technology does not need to be trained beforehand, thus can be utilized on the fly. The class where most of the classifiers agree on is selected as fusion result. The prediction of the output accuracy is not a trivial task, in general the diversity measurements between the involved classifiers must be known [15]. This is not yet considered in this paper, but is an open point for future work. In this experiment, each sensor that is capable of contributing and recognizing the activities of interest (i.e., the four modes of locomotion) is integrated in the ensemble, thus is part of the fusion method.

The experiment session lasted for exactly 14 minutes. By mediating a change in the sensing infrastructure (i.e., in a simulated session, sensors of type *PlayBackSensor* [3] can be turned on and off, thus simulating a (re-)appearance and disappearance) different ensembles were configured. In Table II, the thirteen occurring ensemble configurations are listed together with their IDs, the involved sensor devices

Table II
OVERVIEW OF THE ENSEMBLE CONFIGURATIONS TOGETHER WITH THE CALCULATED ACCURACIES.

ID	Active Ensemble	Accuray
1	RKN	0.604
2	RKN + LLA	0.630
3	RKN + LLA + SHOE	0.779
4	RKN + SHOE	0.488
5	RKN + SHOE + BAC	0.853
6	RKN + SHOE + BAC + LLA	0.840
7	RKN + SHOE + BAC + LLA + RLA	0.846
8	RKN + SHOE + BAC + LLA + RLA + RUA	0.817
9	RKN + SHOE + BAC + LLA + RLA + RUA + LUA	0.870
10	RKN + BAC + LLA + RLA + RUA + LUA	0.818
11	BAC + LLA + RLA + RUA + LUA	0.861
12	BAC + LLA + RLA + LUA	0.852
13	BAC + LLA + RLA	0.820

and the accuracy. This accuracy value was calculated by comparing the predicted class as ensemble output with the actual class that can be achieved from the ground-truth. Before the simulated experimental run with the changes in the sensors' availability was done, each of the thirteen ensembles was configured manually and the specific session with the sensors was executed. This was necessary to gather comparable confusion matrices for each of the listed ensembles with the (approximately) same amount of sensor samples and predicted activities. These confusion matrices are illustrated in Figure 3. Each matrix is assigned with the ID from the ensemble (as listed in Table II), and the resulting accuracy. This accuracy can be easily computed out of the confusion matrices, since the main diagonal indicates the correctly predicted classes, the other values the wrong classified activity labels. Each confusion matrix contains on the x- and y-axis the activity class numbers (i.e., 1=*NULL*, 2=*STAND*, 3=*WALK*, 4=*LIE*, 5=*SIT*). During each run (i.e., 14 minutes) the confusion matrices were filled with approximately 500.000 activity classes (this makes approx. 600 classes per second). This means at each second, 600 comparisons from the predicted label to the actual ground-truth label were done to get a significant matrix as base for the calculation of the accuracy.

After the preparation was done (all *ExperienceItem*s generated, the accuracies of the single sensors and of the ensembles calculated), the experimental session was conducted. Figure 4 presents an overview of the actual accuracy of the configured sensor ensembles for the recognition of the modes of locomotion. The x-axis indicates the time in minutes, but also the ID of the active ensemble (e.g., at minute three, the available sensors were mediated to be *RKN*, *LLA*, and *SHOE*). The y-axis contains the accuracy of the active sensor ensemble. As shown, the accuracy and thus the recognition continuity are robust against changes

 ISBN: 978-1-61208-219-6

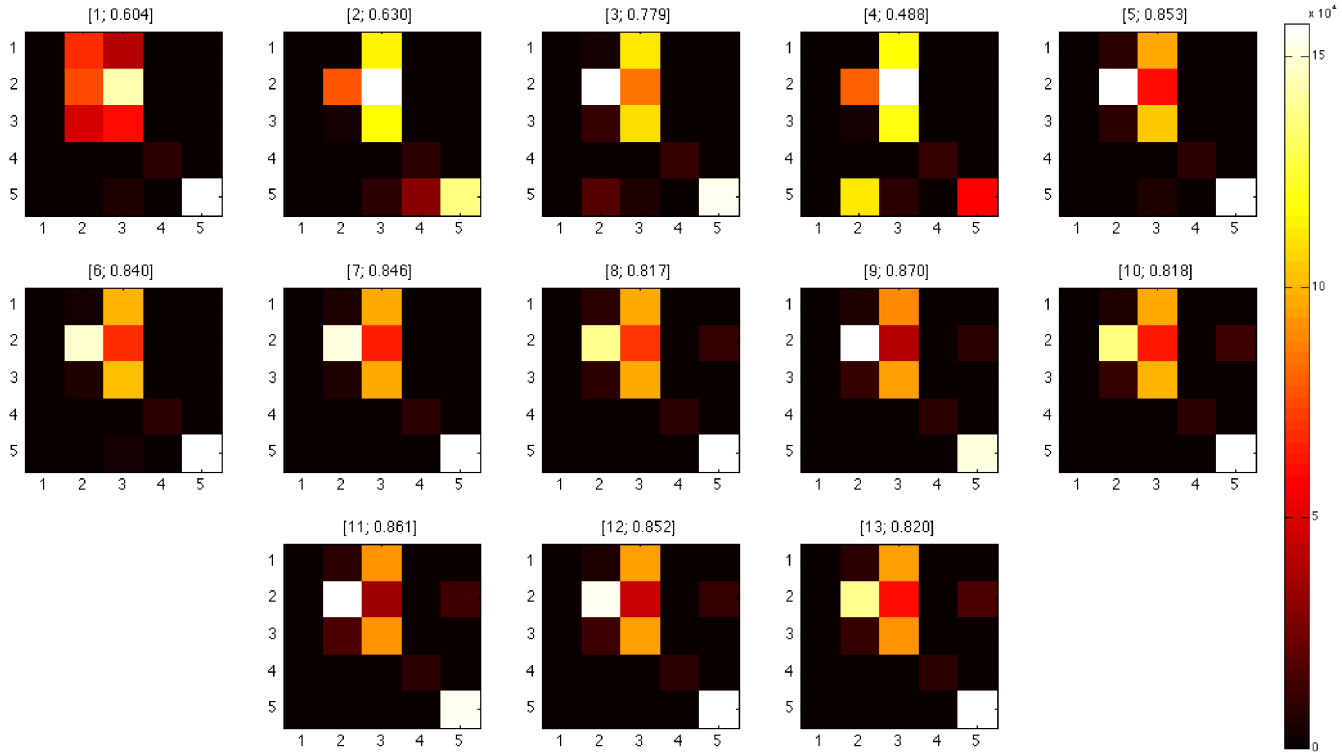


Figure 3. Confusion matrices for the 13 sensor ensembles in the experimental session (the x-axis contains the actual activity class, the y-axis the predicted class). The numbers above each single confusion matrix contain the ensemble ID and the corresponding accuracy compared to the ground-truth.

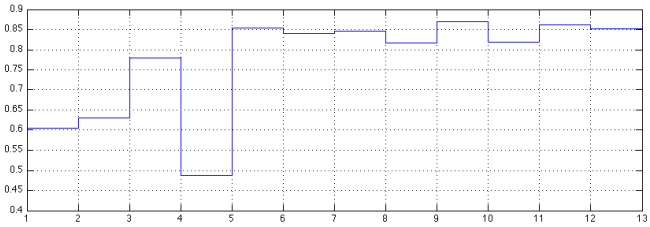


Figure 4. Overview of the accuracies for the occurring ensemble configurations during the experiment session.

in the sensing infrastructure. Based upon the sensor self-descriptions, the system is capable of dynamically adapting to changes in the sensing environment and ensures a continuance of the activity recognition task. This dynamic adaptation to changes in the sensing environment and the resulting stability is a novelty in contrast to conventional activity recognition systems.

V. Conclusion and Future Work

This paper presented the three characteristic application cases *sensor appears*, *sensor reappears*, and *sensor disappears* for an opportunistic activity recognition system. The OPPORTUNITY Framework realizes the concepts of sensor self-descriptions. These XML documents are of highly dynamic nature, since they encode the sensors capabilities in order to recognize certain activities. A sensor can make experience over its life-time (e.g., by manually or autonomous adding of recognition capabilities) and preserve this experience in its self-description in so-called *ExperienceItem*s. Each of these items contains a complete description of an activity recognition chain (i.e., feature extraction, classification and classifier model), together with QoS metrics, like the estimated recognition accuracy. The capability of the OPPORTUNITY Framework to dynamically adapt at runtime to spontaneous changes in the sensing environment is demonstrated in this paper. This enables an accurate, stable and continuous recognition of human activities with dynamically varying sensor settings and can be seen as important building block towards the vision of opportunistic activity and context recognition. This is a novelty in contrast to conventional activity recognition system, since they would fail if sensors disappear, or would not consider new sensors on their appearance.

Concerning future work, one issue is how multiple sensors can be combined to achieve a higher recognition accuracy. The solution so far is to use diversity measures between the sensors. Subsequently, this needs a reliable ground-truth for calculation, which cannot be taken as granted in a real-time application. Nevertheless, to be able to provide an estimation of the accuracy of an ensemble (with the *MajorityVoting* fusion method) these measures are required,

 ISBN: 978-1-61208-219-6

since the danger is that multiple sensors classify the same wrong predicted class, which is then taken as winning fusion result. Currently, work is going to calculate these diversity measures at runtime of the system, and preserving this information additionally in the sensor self-descriptions (at least the diversity measures between two sensors without the comparison to the ground-truth).

ACKNOWLEDGMENT

The project OPPORTUNITY acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 225938.

REFERENCES

[1] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, G. Tröster, H. Sagha, R. Chavarriaga, J. del R. Millán, D. Bannach, K. Kunze, and P. Lukowicz, "The opportunity framework and data processing ecosystem for opportunistic activity and context recognition," *International Journal of Sensors, Wireless Communications and Control, Special Issue on Autonomic and Opportunistic Communications*, vol. 1, December 2011.

[2] D. Roggen, K. Förster, A. Calatroni, T. Holleczek, Y. Fang, G. Troester, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, A. Ferscha, C. Holzmann, A. Riener, R. Chavarriaga, and J. del R. Millán, "Opportunity: Towards opportunistic activity and context recognition systems," in *Proceedings of the 3rd IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2009)*. Kos, Greece: IEEE CS Press, June 2009.

[3] M. Kurz and A. Ferscha, "Sensor abstractions for opportunistic activity and context recognition systems," in *5th European Conference on Smart Sensing and Context (EuroSSC 2010), November 14-16, Passau Germany*, K. K. G. Lukowicz, Paul; Kunze, Ed. Berlin-Heidelberg: Springer LNCS, November 2010, pp. 135–149.

[4] M. Kurz, G. Hölzl, A. Ferscha, A. Calatroni, D. Roggen, and G. Troester, "Real-time transfer and evaluation of activity recognition capabilities in an opportunistic system," in *Third International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE2011), September 25-30, Rome, Italy*, September 2011, pp. 73–78.

[5] M. Kurz, G. Hölzl, A. Ferscha, H. Sagha, J. del R. Millán, and R. Chavarriaga, "Dynamic quantification of activity recognition capabilities in opportunistic systems," in *Fourth Conference on Context Awareness for Proactive Systems: CAPS2011, 15-16 May 2011, Budapest, Hungary*, May 2011.

[6] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, M. Creatura, and J. del R. Millán, "Collecting complex activity data sets in highly rich networked sensor environments," in *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS), Kassel, Germany*. IEEE Computer Society Press, June 2010.

[7] J. Mantyjarvi, J. Himberg, and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 2, 2001, pp. 747 –752 vol.2.

[8] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, A. Ferscha and F. Mattern, Eds. Springer Berlin / Heidelberg, 2004, vol. 3001, pp. 1–17.

[9] N. Ravi, D. Nikhil, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence(IAAI*, 2005, pp. 1541–1546.

[10] D. Minnen and T. Starner, "Recognizing and discovering human actions from on-body sensor data," in *In Proc. of the IEEE International Conference on Multimedia and Expo*, 2005, pp. 1545–1548.

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, pp. 74–82, March 2011.

[12] A. Calatroni, D. Roggen, and G. Tröster, "Automatic transfer of activity recognition capabilities between body-worn motion sensors: Training newcomers to recognize locomotion," in *Eighth International Conference on Networked Sensing Systems (INSS'11)*, Penghu, Taiwan, Jun. 2011.

[13] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, R. Verdone, Ed. Springer Berlin / Heidelberg, 2008, vol. 4913, pp. 17–33.

[14] C. Lombriser, N. B. Bharatula, D. Roggen, and G. Tröster, "On-body activity recognition in a dynamic sensor network," in *Proceedings of the ICST 2nd international conference on Body area networks*, ser. BodyNets '07. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 17:1–17:6.

[15] R. Chavarriaga, H. Sagha, and J. del R Millan, "Ensemble creation and reconfiguration for activity recognition: An information theoretic approach," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, oct. 2011, pp. 2761 –2766.

[16] C. Villalonga, D. Roggen, and G. Tröster, "Shaping sensor node ensembles according to their recognition performance within a planning-based context framework," in *Eighth International Conference on Networked Sensing Systems (INSS'11)*, 2011.

[17] M. Botts and A. Robin, "OpenGIS Sensor Model Language (SensorML) Implementation Specification," OGC, Tech. Rep., Jul. 2007.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley - InterScience, 2001.

 ISBN: 978-1-61208-219-6

Adaptation Algorithm for Navigation Support in User Adaptive Enterprise Application

Inese Supulniece
Institute of Information Technology, Riga Technical University
Kalku 1, Riga, Latvia, LV-1658
Inese.Supulniece@rtu.lv

***Abstract*—User Adaptive Enterprise Application supports users in identification of more efficient variations of business process executions. It is the set of adaptive components to be added to standard enterprise application. Adaptive navigation support is one of identified components with the aim to help users execute routine activities faster, reduce amount of mistakes and support new users of the system. The paper presents a meta-model, architecture and adaptation algorithm behind the adaptive navigation support. Business process constraints are used to describe business rules and restrictions. Process execution patterns are used to discover characteristics and preferences of individual users. The proposed algorithm is evaluated using simplified sales process simulation in Microsoft Dynamics AX and task management process simulation. The results of the early evaluation show that adaptive navigation component supports business rules and individual variations of business process execution. It also indicated some limitations of applying business process constraints on user interface level.**

Keywords-user adaptive system; enterprise application; adaptation algorithm; recommendation.

I. INTRODUCTION

Today, in rapidly changing environment, business processes are dynamic [1]. The need to adapt a process has been a topic of interest in the recent years [1]. Enterprise applications are used to execute business processes. Usually, these are packaged applications providing standardized implementations of business processes. Users of enterprise applications either use predefined workflows or rely on user documentation and best practices to execute their business processes [2]. Besides these standard capabilities, in many cases, users also can use other functions provided by enterprise applications subject to their access rights. That means that users have possibilities to introduce their own variations in process execution. By considering these variations, users might come up with more efficient ways of executing business processes [3]. If an enterprise application supports users in identification of more efficient variations of business process execution and enables for continuous execution refinement it is referred as to as User Adaptive Enterprise Application (UAEA) [4].

There exist various approaches, on how to manage business process variants without violating organisational rules. One of them is description of business processes, using process constraints [5]. Business process constraints can express minimal restrictions on the selection and ordering of tasks of the targeted business process, thus providing a degree of flexibility in process execution. Constraint-based models are considered to be more flexible than traditional models because of their semantics: everything that does not violate constraints is allowed [5].

The objective of this paper is to present the meta-model, architecture and adaptation algorithm behind Adaptive Navigation Support (ANS) of UAEA. This component is using 1) business process constraints to keep main rules of the processes under control while allowing different business process execution variants; and 2) task execution patterns to manage individual user oriented process variants.

The rest of paper is structured as follows: Section 2 provides brief introduction to UAEA. Section 3 presents adaptation constraints for business process variants. The ANS component is explored in Section 4. The paper concludes with Section 5, where conclusion and further research are discussed.

II. USER ADAPTIVE ENTERPRISE APPLICATION

There are multiple ways the enterprise applications could be adapted, e.g. [1], [6], [7]. In the context of UAEA, the adaptation engine generates the user-oriented view of business processes in the enterprise application. Given that ERP systems are mainly used for repetitive tasks [8], the user-oriented process adaptation uses previously observed users' behavior to optimize performance of business activities. [6] and [7] discusses the same problems and similar approaches for solving them, however, architecture and logics differ per each research (also for this paper). Each of proposals has its own motivating business case, benefits and restrictions, thus it is rather impossible to compare their effectiveness. For example, the adaptation mechanism in [6] applies two data sets: the process model, which describes business rules and the sequence graph, which comprises nodes representing the individual process steps. An directed edge between two nodes A and B of the sequence graph describes a temporal sequence that process step B follows immediately after A and edge value represents the likehood of following a particular path through the process. Our adaptation mechanism uses business process constraints to describe business rules and an ordinary sequence to keep individual process execution variants. The choise between usage of the full business process model (as in [6]) or business process constraints (as in this paper) depends on the flexibility degree of the process.

 ISBN: 978-1-61208-219-6

The idea of the UAEA and main characteristics are described in [4], where the model of UAEA is elaborated. The overall goal of the UAEA is to identify possibilities of existing enterprise applications to raise performance efficiency (see Figure 1). Related operational goals are: optimization of routine activities, preventing mistakes, decreasing the learning time for new processes and for new employees. Technically, system should optimize routine activities, prevent mistakes and support non-routine activities. This is measured in process execution time and amount of mistakes.

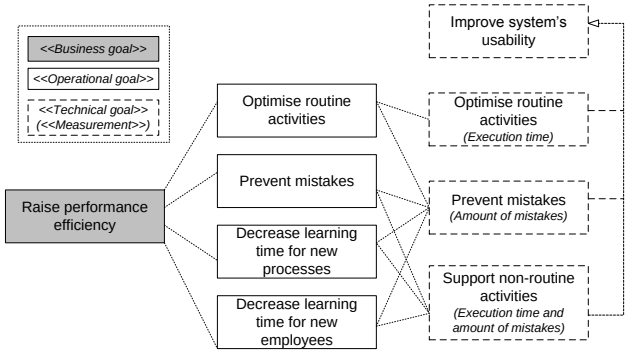


Figure 1. The goal model of the UAEA.

UAEA is the set of six adaptive components to be added to a standard enterprise application:

Adaptive process execution overview shows full process or part of the process, current activity and possible paths to finish the process. The aim of this recommendation is to provide local or global guidance for user, especially for non-routine activities.

Adaptive navigation support (ANS) presents recommendation block with recommended navigation item, mandatory and prohibited activities for particular process. The aim of this recommendation is to help user execute activities faster, to reduce amount of mistakes and support new users of the system.

Adaptive information support recommends related documents, systems or data based on local or global patterns.

Adaptive decision support recommends possible decisions based on local or global decision patterns.

Adaptive problem preventing presents most common problems and solutions related to current activity. It prevents possible mistakes for non-routine activities or new users.

Adaptive error and exception handling notifies user about incompleteness in process execution, e.g., missed activity or not finished process.

Idea of the ANS for the UAEA lies in the following observation [9]: users use enterprise application to accomplish their tasks, usually consisting of multiple steps; each user or user group has a preferred sequence of the steps (task execution patterns). UAEA attempts to exploit such usage patterns with the aim to improve performance efficiency.

This paper explores a meta-model, architecture and adaptation algorithm behind ANS component.

III. ADAPTATION CONSTRAINTS FOR BUSINESS PROCESS VARIANTS

In large enterprises, it can be observed that a common business processes exists in many variations across different parts of the organisation [10]. When supporting business processes there is a difficult trade-off to be made between control and flexibility [5].

Control is achieved with restrictions for the process adaptation, which are modeled as rules or constraints. Business process constraints are suitable for supporting flexible processes that allow many different executions [5]. Most theoretical process modeling languages, such as Petri Nets, process algebras, BPMN, UML and EPCs define direct causal relationships between activities in process models. Opposed to this, constraint-based languages are of a less procedural nature and use a more declarative style [5]. Declarative languages are more flexible by nature, and it is more likely that users working in such an environment need support, e.g. recommendations [7].

There have been proposed a number of constraint languages in various disciplines, e.g., ConDec [11], Object Constraint Language [12], MiniZinc [13]. These are extensive approaches; consequently they require specific knowledge and complex algorithms for run-time process adaptation based on available constraints.

Lu et al. [14] presents how task selection constraints can be specified at design time, through selection constraints. This approach was adapted for the ANS, because it is unsophisticated, efforts for managing and using the business process constraints should be kept minimal and seems to be promising approach for combining process constraints and task executions patterns in adaptation algorithm.

In [14], the following classes of selection constraints have been identified:

(1) Mandatory constraint *man* defines a set of tasks that *must be executed* in every process variant, in order to guarantee that intended process goals will be met.
(2) Prohibitive constraint *pro* defines a set of tasks that *should not be executed* in any process variant.
(3) Cardinality constraint specifies the minimal *minselect* and maximal *maxselect* cardinality for selection among the set of available tasks.
(4) Inclusion constraint *inc* expresses the dependency between two tasks T_x and T_y, such that the of T_x imposes restriction that T_y must also be included. Prerequisite constraint *pre* is the inverse of an inclusion constraint.
(5) Exclusion constraint *exc* prohibits T_y from being included in the process variant when the T_x is selected.
(6) Substitution constraint *sub* defines that if T_x is not selected, then T_y must be selected to compensate the absence of the former.
(7) Corequisite constraint *cor* expresses a stronger restriction in that either both T_x and T_y are selected, or none of them can be selected, i.e., it is not possible to select one task without the other.

 ISBN: 978-1-61208-219-6

(8) Exclusive-Choice constraint *xco* is also a more restrictive constraint on the selection of alternative tasks, which requires at most one task to be selected from a pair of tasks (T_x, T_y).

The mentioned classes of selection constraints are re-used in the ANS component.

IV. ADAPTIVE NAVIGATION SUPPORT

The main goal of the ANS component is to optimize routine activities, prevent mistakes and also support new users during non-routine activities. The changing object for this component is user and task. It means that adaptation result differs per user and takes into account situational aspects.

Figure 2 presents a meta-model of the ANS component, which illustrates the main concepts used by the adaptation algorithm. An enterprise application consists of *user interface (UI) elements*, which are mapped to the *activities* of the process. Capturing the activities, which are not related to the control UI elements of the application, is out of the scope of this research and proposed adaptation algorithm. Each UI element belongs to some UI *form* or *window*. *Constraint* consists of two activities. Constraints are defined separately for each form/window. *Process execution pattern* comprises activities representing the actually executed process steps. It consists of two or more activities and it is related to the *user*, who executed the particular pattern. Each pattern has the attribute – *frequency of execution* – how many times the pattern was executed. The set of *global patterns* include all execution patterns, despite the user, who created it. The set of *local patterns* include only those patterns, which were executed by the particular user.

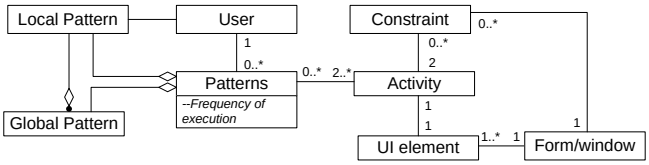


Figure 2. The meta-model of the ANS component.

The architecture of the ANS component is illustrated in Figure 3. It consists of data bases (event logs); repositories (users, constraints, activities, execution patterns); engine for the adaptation algorithm and the user interface of the ANS. Types of business rules or constraints are adapted from [14] and are available in the form:

<form/window>,<constraint_type>{Tx, Ty}.

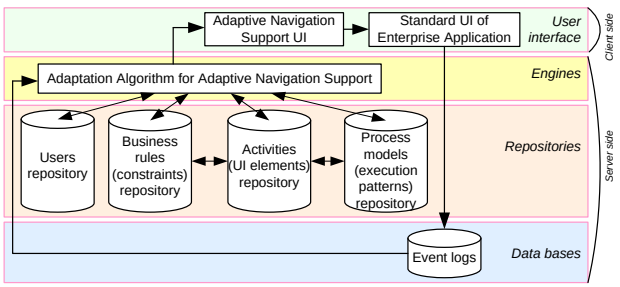


Figure 3. The architecture of the ANS component.

Process models or execution patterns are saved as the sequences of activities $a_1, a_2, \ldots a_n$. All activities executed by each individual user are perceived and stored as business process patterns.

A. Description of the adaptation algorithm

The current activity, process execution patterns (individual and global) and business process constraints forms the input to the adaptation algorithm (see Figure 4). The main output is recommended next step.

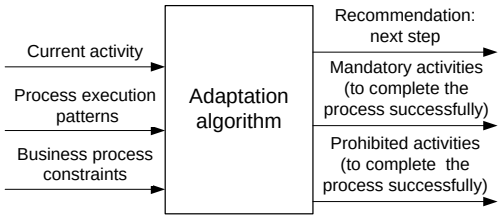


Figure 4. Input/ouput view of the adaptation algorithm.

To realize the adaptation algorithm of the ANS, the following data sets are introduced: *1) M* – consists of activities $M_1, M_2, \ldots, M_k$, which are mandatory; *2) E* – consists of activities $E_1, E_2, \ldots E_u$, which are prohibited; 3) *I* – consists of executed activities $I_1, I_2, \ldots I_p$. All mentioned data sets are sequences.

Figure 5 presents simplified view of the adaptation algorithm behind the ANS. Firstly, the system reads the activity *A* performed by the user, identifies the form F_o and selects all constraints, which include activity *A*. When the user executes any activity inside some form/window F_o, then all activities from the set of constraints $\{F_o, man\{T_i\}\}$ are automatically added to the set *M* and all activities from the

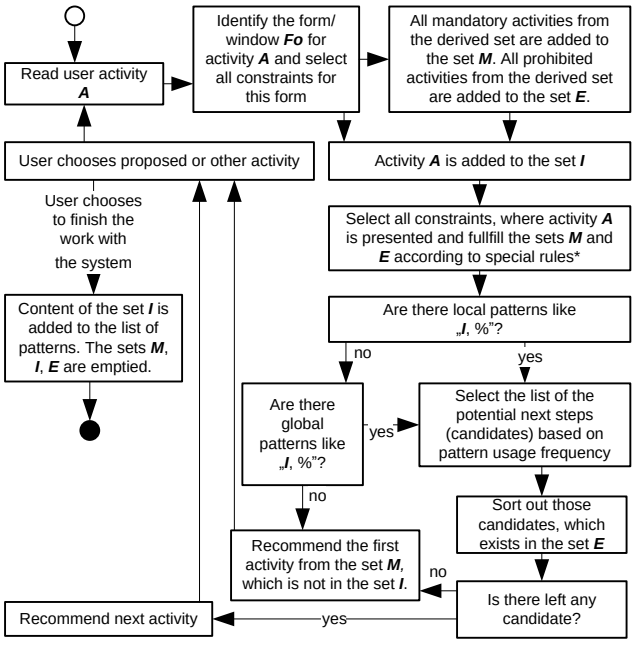


Figure 5. Simplified view of the adaptation algorithm behind the ANS.

TABLE I. THE LIST OF SPECIAL RULES

When any element H is added to the set M, then following rules should be verified:	
If there exists constraint	***Then execute following action***
$\{F_o, inc(H, Ty_i)\}$	Ty_i *is added to the set M*
$\{F_o, exc(H, Ty_i)\}$	Ty_i *is added to the set E*
$\{F_o, pre(Tx_i, H)\}$	Tx_i *is added to the set M*
$\{F_o, cor(H, Ty_i)\}$	Ty_i *is added to the set M*
When any element H is added to the set E, then following rules should be verified:	
If there exists constraint	***Then execute following action***
$\{F_o, sub(H, Ty_i)\}$	Ty_i *is added to the set M*
$\{F_o, cor(H, Ty_i)\}$	Ty_i *is added to the set E*

set $\{F_o, pro\{T_i\}\}$ are automatically added to the set *E*. But activity *A* is added to the set *I*.

Secondly, all constraints (including activity *A*) are reviewed by the system using special rules and the sets M and E are supplemented. For example, if there exists constraint $\{F_o, inc(A, Ty_i)\}$, then both activities *A* and Ty_i must be executed together. Consequently *Tyi* is added to the set of mandatory activities *M*. The special rules are listed in Table 1.

Further the list of local patterns is identified. If there are not local patterns, then system looks for global patterns. The list of the potential next steps is prepared according to the pattern usage frequency. If candidate exists in the set E, then it is removed from the list of the potential next steps. The system recommends the candidate with the highest pattern usage frequency index. If there are no candidates, then system recommends the first element from the set M, which is not executed yet. Next, it is up to user to utilize or ignore the recommendation.

B. Initial testing of the algorithm

The aim of initial testing was to prove: (1) if constraint types from [13] can be applied on user interface level and (2) if logic of the algorithm provides expected results.

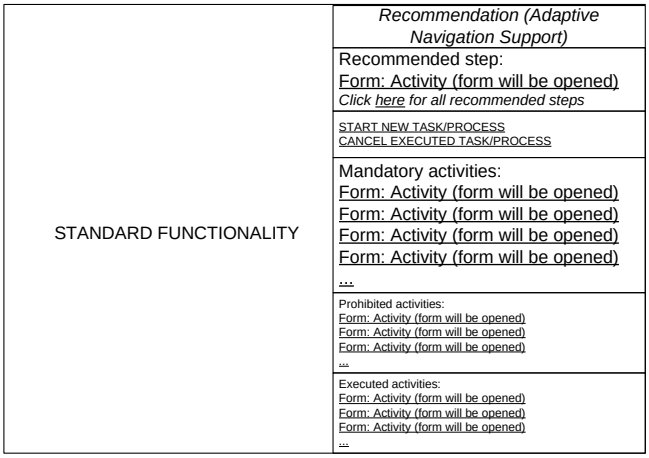


Figure 6. User interface prototype of the ANS.

Testing was performed as 1) simplified task management process simulation and 2) simplified sales process simulation in Microsoft Dynamics AX [15] Sales module and system proposed recommendations according to the user interface prototype, which is presented in Figure 6.

At the current stage of the research the main idea of testing was to perform initial validation of logics. Usability, performance and effectiveness testing is planned in nearest future.

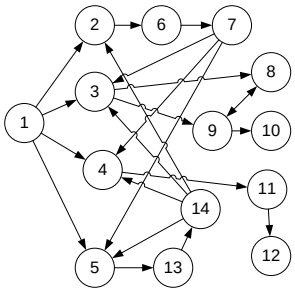
The results of the testing indicated limitations and problems of applying mentioned constraints on user interface level.

1) Task management process support

Before the testing, the following preparation works were done:

- Business process constraints were transferred to user interface level – see Table II.
- Four different process execution alternatives were stored in execution patterns repository – see Table III.

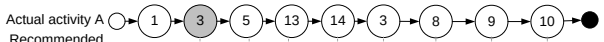
In the task management process, a user selects new or existing task. The task can be completed, forwarded, closed and/or supplemented with additional information. The possible process execution alternatives are illustrated in Figure 7.



No	Form	Activity
1	Main form	Proceed the task
2	New/existing task	Complete the task
3	New/existing task	Forward
4	New/existing task	Close the task
5	New/existing task	Additional information
6	Solution form	Add comment
7	Solution form	Save
8	Forwarding form	Select department
9	Forwarding form	Select employee
10	Forwarding form	Save
11	Closing form	Add comment
12	Closing form	Save
13	Additional information	Add comment
14	Additional information	Save

Figure 7. Task management: variations in process execution.

One variant of the process execution was simulated during the initial testing. The simulated process included nine activities. Seven activities corresponded to system recommendations and one recommendation failed. Testing report is available in Figure 8.



Recommended activity	2	5	13	14	3	8	9	10	-
M	1	1,5	1,5	1,5,13	1,5, 13,14	1,5,13, 14,3	1,5,13, 14,3,8	1,5,13, 14,3,8,9	1,5,13,14, 3,8,9,10
E	-	-	-	-	-	2,4	2,4	2,4	2,4
I	1	1	1,5	1,5,13	1,5, 13,14	1,5,13, 14,3	1,5,13, 14,3,8	1,5,13, 14,3,8,9	1,5,13,14, 3,8,9,10

Figure 8. Task management: testing results.

2) Sales process support

The three alternatives of the basic sales process activities were executed during the testing. The first alternative was linked to the user 1 – very careful person, who verifies the data before doing any action, e.g., prove the stock before adding the product to the offer. Second alternative was linked to the user 2 - person, who trusts the system and does only basic steps. Third alternative was linked to the user 3.

TABLE II. BUSINESS CONSTRAINTS

Type	*Form*	*Activities*
EXC	*New/existing task*	*3,2*
EXC	*New/existing task*	*3,4*
PRE	*New/existing task*	*5,3*
PRE	*Forwarding form*	*8,9*
PRE	*Additional information*	*13,14*
PRE	*Closing form*	*11,12*
PRE	*Solution form*	*6,7*

TABLE III. BUSINESS PROCESS PATTERNS

Pattern	*Usage frequency*	*User*
1,2,6,7	*5*	*1*
1,3,8,9,10	*2*	*1*
1,4,11,12	*3*	*1*
1,5,13,14,3,8,9,10	*1*	*1*

These alternatives were stored as process execution patterns and business process constraints were transferred to user interface level, e.g., delivery address and currency is mandatory information.

C. Limitations

The main problem is related to usage of constraints, because originally constraints in this form were developed for description of business processes. Constraints in current form define only relations between every 2 activities. For example, none of described constraints allows specifying the following rule: if *client is selected*, then afterwards it is mandatory to *Save* the form OR *Cancel* the form. One option would be to write this rule as *inc*{(client is selected), *xco*{Save, Cancel}}. But this requires more sophisticated algorithm, which might end with performance issues on real life system and data amount.

Another problem is related to user interface design of described component. How to track read-only fields; when user uses it; when they stop to be relevant to particular activity?

Consequently, currently design of Adaptive Navigation Support component recommends only next executable activity and opens the full form, where it is located.

V. CONCLUSION AND FUTURE RESEARCH

This paper presented a meta-model, architecture and adaptation algorithm behind adaptive navigation support component in user-adaptive enterprise application. Business process constraints are used to describe business rules and restrictions. Process execution patterns are used to discover characteristics and preferences of individual users.

Important problems are identified at current stage, e.g., limitations of existing form of defining the constraints. Now the aim is to develop an interactive prototype of the Adaptive Navigation Support component and test usability, effectiveness and performance by real users.

Also valuable ideas rose during the research, e.g. differentiation between mandatory and optional constraints as suggested by [5].

ACKNOWLEDGMENT

This work has been supported by the European Social Fund within the project «Support for the implementation of doctoral studies at Riga Technical University».

REFERENCES

[1] G. Hermosillo, L. Seinturier, L. Duchien, „Using Complex Event Processing for Dynamic Business Process Adaptation", In Proc. of the 7th IEEE 2010 International Conference on Services Computing, 2010. DOI : 10.1109/SCC.2010.48

[2] T.A. Curran, A. Ladd, "SAP R/3 Business Blueprint: Understanding Enterprise Supply Chain", Prentice Hall PTR, Upper Saddle River, 2000.

[3] H. Topi, W. Lucas, T. Babaian, "Identifying usability issues with an ERP implementation", In Proc. of ICEIS 2005, pp. 128-133.

[4] I.Supulniece, J.Grabis, "Modeling of user adaptive enterprise applications", In Proc. of ICEIS 2012, in press.

[5] M. Pesic, M.H. Schonenberg, N. Sidorova, W.M.P. van der Aalst, "Constraint based workflow models: Change made easy". In Proc. of OTM Confederated International Conferences 2007, 2007, pp. 77-94.

[6] C. Dorn, T. Burkhart, D. Werth, S. Dustdar, „Self-adjusting recommendations for peopledriven ad-hoc processes", In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, Springer, Heidelberg, 2010, pp. 327–342.

[7] B. Weber, B.F. van Dongen, M. Pesic, C.W. Guenther, W.M.P. van der Aalst, „Supporting flexible processes through recommendations based on history", Eindhoven University of Technology Eindhoven, BETA Working Paper Series, 2007. ISBN: 978-90-386-1038-2.

[8] H. Klaus, M. Rosemann, G.G. Gable, "What is ERP?" Information Systems Frontiers 2:2, 2000, pp. 141-162.

[9] I. Supulniece, J. Grabis, "Discovery of personalized information systems usage patterns", In Proceedings of ICIST, Kaunas, Lithuania, 2010, pp. 25-32.

[10] M. Weidlich and M. Weske. "Structural and behavioural commonalities of process variants". In Proc. of ZEUS'10, Berlin, Germany, CEUR vol.563, 2010, pp. 41-48, CEUR-WS.org

[11] M. Pesic and W.M.P. van der Aalst, "A declarative approach for flexible business processes". In J.Eder and S.Dustdar, (eds.), Business Process Management Workshops, Workshop on Dynamic Process Management (DPM 2006),LNCS, vol. 4103, Springer-Verlag, Berlin, 2006, pp. 169-180.

[12] Object Management Group, "Object constraint language specification version 2.3.1". OMG, 2012

[13] N. Nethercote, P.J. Stuckey, R. Becket, S. Brand, G.J. Duck and G. Tack, "MiniZinc: Towards a standard CP modelling language". In CP, LNCS 4741, 2007, pp. 529–543

[14] R. Lu, S. Sadiq, G. Governatori, X.Yang, "Defining adaptation constraints for business process variants", In Proc. of BIS, 2009, pp. 145-156

[15] Microsoft Dynamics AX, Retrieved from http://www.microsoft.com/en-us/dynamics/erp-ax-overview.aspx

 ISBN: 978-1-61208-219-6

A QoS Optimization Model for Service Composition

Silvana De Gyvés Avila, Karim Djemame
School of Computing
University of Leeds
Leeds, UK
e-mail: {scsdga, scskd}@leeds.ac.uk

***Abstract*— The dynamic nature of the Web service execution environment generates frequent variations in the Quality of Service offered to the consumers, therefore, obtaining the expected results while running a composite service is not guaranteed. Adaptation approaches aim to maintain functional and quality levels, by dynamically adapting composite services to the environment conditions reducing human intervention. This paper presents an adaptation approach based on self-optimization. The proposed optimization model performs service selection based on the analysis of historical and real QoS data, gathered at different stages during the execution of composite services and the establishment of priorities between the service quality attributes. Experimental results show significant improvements in the global QoS of the use case scenario, providing reductions up to 16% in the global cost and 14% in response time.**

Keywords - Web service composition; adaptation; optimization; Quality of Service.

I. INTRODUCTION

Web services are modular, self-contained and reusable software components that rely on open XML-based standards to support machine-machine interactions over distributed environments [1]. Some of the benefits offered by services include time/cost reduction during software development and maintenance. When a single service does not accomplish a consumer's requirement, different services can be used in conjunction to create a new value-added service to fulfil this requirement. A composite service provides a new software solution with specific functionalities and can be seen as an atomic component in other service compositions or as a final solution to be used by a consumer [2]. The process of developing a composite Web service is called service composition.

Development in the field of service composition has resulted in a set of dataflow models (orchestration and choreography), approaches (static, dynamic, manual and automatic) and techniques (model-driven, declarative, workflow-based, ontology-driven and AI-Planning) that enable composition from different perspectives. However, some challenges still remain open, which are closely related to automatic-dynamic service composition and include the implementation of mechanisms that enable: Quality of Service awareness, adaptive capabilities, risk awareness, conformance, security and interoperability.

The approach proposed in this paper is mainly focused on adaptive mechanisms for service composition. Adaptive mechanisms provide software systems with capabilities to self-heal, self-configure, self-optimize, self-protect, etc., considering the objectives the system should achieve, the causes of adaptation, the system reaction towards change and the impact of adaptation upon the system [3].

Adaptation in service composition aims to mitigate the impact of unexpected events that take place during the execution of composite services, maintaining functional and Quality of Service (QoS) levels. By implementing adaptive mechanisms, composite services should be able to morph and function in spite of external and internal changes, searching to maximize the composition potential and reducing as much as possible human involvement.

This work presents a self-optimization solution for service composition. The proposed optimization model performs service selection based on historical QoS data and real data, which is collected at runtime during different stages of the composite service execution. Upon invocation, a set of tasks are executed as defined in the service workflow. QoS data evaluation from previous tasks enables the model to determine priorities between the QoS attributes, and these priorities are applied during service selection. The approach has been implemented in a framework and was evaluated empirically by analyzing the execution through a use case. The major contribution of this paper is:

- The optimization model for service composition that analyzes global QoS from previous tasks in order to determine priorities for service selection.

This paper is structured as follows: background and related work are described in Section II. Section III presents the proposed framework, service selection and optimization models. Section IV presents the experimental description and results. Conclusions and future work are given in Section V.

II. BACKGROUND AND RELATED WORK

In service composition, it is necessary to have a set of available services that offer certain functionality and also fulfil Quality of Service constraints [4].

QoS properties refer to non-functional aspects of Web services, such as performance, reliability, scalability, availability and security [5]. By evaluating the QoS aspects of a set of Web services that share the same goals, a

 ISBN: 978-1-61208-219-6

consumer could identify which service meets the quality requirements of the request.

The QoS attributes of a service can be evaluated during design and execution time. At design time, these attributes help in order to build a composite service based on the QoS requirements of the user. While at execution time, they can be monitored to maintain the desired QoS level. Information about these attributes can be obtained from the service's profile [6], nevertheless, when this information is not available, it can be obtained by analyzing data collected from past invocations [7].

Different approaches have been presented to evaluate QoS attributes in service composition, aiming to select a set of components that optimize the global QoS. Some of these approaches are based on the works described in [7] and [8], which proposed mathematical models to compute QoS of composite services based on the QoS of their components and consider time, cost, reliability, availability and reputation as the quality criteria to evaluate.

To experience an expected behaviour during the execution of a composite service, it is important to consider the QoS aspects of the services involved, as their drawbacks will be inherited by the composite service. However, unexpected events occur, e.g., services become unavailable or exhibit discrepancies in their QoS [9], bringing the need of mechanisms such as adaptation, in order to restore and maintain the functional and quality aspects of the composition.

Based on the objectives of the composition and the causes and impact of adaptation, different self-adaptive properties can be selected and implemented. The most used properties in service composition approaches are self-healing [10], self-configuration [11] and self-optimization [12]. Each of these properties can be related to different attributes, like availability, survivability, maintainability, reliability, efficiency and functionality [13].

Self-healing mechanisms aim to prevent composite services from failing, from functional and non-functional perspectives. Projects such as those are presented in [14-21], apply self-healing approaches, where new services are selected and invoked after a functional failure or a QoS constraint violation.

In self-configuring approaches, like those presented in [9] and [22], service selection is performed by searching for an optimal configuration of components based upon the initial constraints.

On the other hand, mechanisms that implement self-optimization are closely related to the selection of services at runtime, in order to maintain the expected QoS of the entire composition. Examples of works belonging to this category are described in [16], [21] and [22].

Although these approaches are closely related with the work described in this paper, there are meaningful differences. Firstly, the proposed optimization approach takes into consideration the QoS values measured from previous tasks at the time of selecting a new service. Secondly, optimization of QoS is also considered when the measured QoS values at certain point of the composite service execution is better than expected, enabling the improvement of other QoS attributes.

III. SYSTEM MODEL

The implementation and evaluation of the proposed approach requires to setup an environment in which QoS aware and adaptive composition can be executed. The system model illustrated in Fig. 1 has been developed with this purpose. Its core components are described as follows:

- Service Binder: binds dynamically each of the tasks in the composition to executable services. These services are selected using functional and QoS criteria.
- Service Selector: by using required functional and quality information, this module searches in the service registry for those elements that fulfil functional and quality requirements.
- Predictor: obtains estimates for the QoS attributes of the selected services by using predictive algorithms and a collection of historical QoS data.
- Sensors: collect information about different events at run time and send it to the adaptation module. Events are related to quality aspects of the involved compositions' elements.
- Adaptation module: monitors and analyzes the behaviour of composite services at runtime and according to its analysis, determines when it is needed to perform certain changes in order to improve/maintain the offered QoS of the compositions.
- Effectors: apply the actions provided by the adaptation module, enabling composite services to adapt at runtime.
- Composition engine: executes the composite services (processes' definitions).

Composite services are considered to consist of a series of abstract tasks that will be linked to executable services at runtime. To obtain these services, for each task the service binder invokes the service selector (SS) and it requests the desired characteristics that the component service should provide.

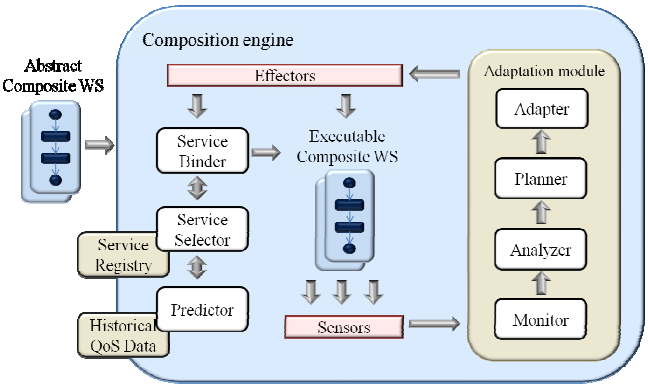


Figure 1. System model.

The SS performs a search into the service registry based on the provided functional requirements. For each of the pre-selected services (candidates), the SS module invokes the predictor to obtain its estimated QoS. The SS compares the results and sends the information about the service that suits the request to the binder.

When the composite service is being executed, sensors capture information about the behaviour of the service and its components, QoS data is being stored in the historical database. Sensors send this information to the adaptation module, which determines if adaptation is needed and the appropriate adaptation strategy. Finally, it sends the actions to be performed to the corresponding effectors, in order to maintain/improve the QoS of the composition.

It is considered that at the time of invoking a composite service, the system has available data from previous executions of the different possible components, in order to obtain accurate predictions about these components' quality characteristics. Also, for each task of the composite service, there exist at least two concrete services to invoke.

A. Service Selection Model

Different QoS attributes can be associated with Web services [7-8], which could be used as a differentiating point in the preference of consumers. In this work, the following quality attributes, which have been used in other approaches ([4],[14-16]), will be considered for each service:

- Response time: the time consumed between the invocation and completion of the service operation [14];
- Cost: fee charged to the consumer when invoking a service [16].

Estimation of QoS values is a key step during service selection process. Estimated values are calculated using historical QoS data recorded from previous executions. This data is filtered, discarding values considered as outliers and the average of the last N executions of the remaining subset is obtained.

Concrete services are searched in the registry by name, assuming that this parameter includes/describes the service's functionality. The resulting set of candidate services is sorted according to the relationship between their estimated response time and cost. Due to these attributes having different units of measure, the raw values are first normalized with natural logarithms. Results are then computed using the Simple Additive Weighting formula:

$$W_i = t_i\,(w_1) + c_i\,(w_2) \quad (1)$$

where:
t_i corresponds to the service estimated response time,
c_i corresponds to the service estimated cost,
w_1 and w_2 correspond to weights where $w_1 + w_2 = 1$ and $w_1, w_2 \leq 1$.

Input:
estRT → estimated accumulated response time
estC →estimated accumulated cost
rt → real response time
rc → real cost
w_1, w_2 → weights
ω→ maximum difference between estRT and rt
ϕ → maximum difference between estC and rc

Output:
α → response time weight
β →cost weight

(1) ψ ← **calculate** response time difference (estRT - rt)
(2) δ ← **calculate** cost difference (estC - rc)
(3) α ← β ← 0.5
(4) Sort by response time
(5) **if** ψ ≥ ω || -δ ≥ ϕ **then**
(6) α ← w_1
(7) β ← w_2
(8) **else**
(9) Sort by cost
(10) **if** δ ≥ ϕ || -ψ ≥ ω **then**
(11) α ← w_2
(12) β ← w_1
(13) **return** α and β

Figure 2. QoS evaluation algorithm.

B. Optimization Model

Monitoring the execution of services is a critical task in the adaptation process. By monitoring and collecting data from services executions, based on their behaviour it is possible to take decisions about future actions [23]. As part of this work, at runtime QoS information is collected from service, task and process perspectives, where service corresponds to concrete Web services; task to elements within the composite service that invoke services; and process to the entire composition (service workflow). Response time is measured during each stage of the process, while cost is obtained from the WSDL files of the services. The QoS values of a task are registered as an individual invocation and as the accumulated QoS of the composition at the time of executing the task.

The optimization approach is based on the service selection model previously described. It uses variable weights and performs a service reselection on the obtained set of candidates. When the accumulated response time (or cost) of the previous activity in the process is less than expected, it provides some slack that can be used while selecting the next service in the process.

Before invoking a Web service operation, the measured accumulated QoS values of the previous task are evaluated and compared to the corresponding estimated values. The algorithm presented in Fig. 2 describes the QoS evaluation method applied during optimization. After obtaining the differences between the estimated and real QoS values (steps 1 and 2), these values are compared to the maximum desired percentage of difference between real and estimated values, represented by ω and ϕ. The first comparison is performed based on response time (step 5), if there is no

 ISBN: 978-1-61208-219-6

adaptation required, then evaluation is carried out based on cost (step 10). The algorithm returns α and β (step 13), which are the new weights to apply in the service selection process. These weights are established as float values that give priority to a certain attribute.

IV. EVALUATION

In order to asses the effectiveness of the proposed optimization approach, an experimental environment was setup and a composite service was developed as use case.

Elements described in Section III were deployed and configured within the experimental environment. Experiments were carried out to address the following question:

- Is there any improvement in the global QoS when using variable weights during service selection as part of a self-optimization mechanism?

A. Experimental Environment

The experimental environment is illustrated in Fig. 3. It consists of one computer with Windows Vista, 4GB RAM and one Intel core2 duo 2.1GHz processor (node 1); and two virtual machines with lubuntu 11.10, 512 Mb RAM and one processor (node 2 and 3). Node 1 hosts the BPEL engine (Apache ODE 1.3.4), service registry (jUDDI 3.0.4), historical data base (MySQL 5.1.51) and one application server (Tomcat 6.0.26). Node 2 and 3, host one application server each (Tomcat 6.0.35). Web services, are allocated in the application servers.

This environment works in a Local Area Network (LAN), and considers response time of Web services running over a Wide Area Network (WAN) when executing the local services. However, in further experiments it is important to perform a detailed analysis of the behaviour of Web services (e.g., faults, availability, latency) over a WAN, in order to obtain results closer to a realistic scenario.

B. Experiment Description

The test case is a BPEL [24] service that implements a travel planning process. It validates a credit card, performs flight and hotel reservations in parallel, and finally invokes a car rental operation. This service is hosted and invoked from Node 1.

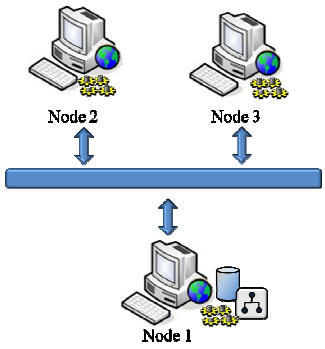


Figure 3. Experimental environment.

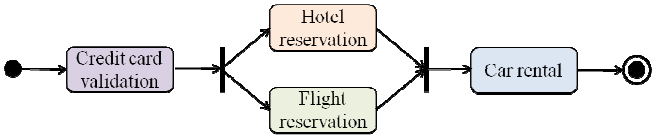


Figure 4. Travel planning process.

The travel planning service is illustrated in Fig. 4. Per each of the tasks in the process, there are 9 candidate services that fulfil the required functionality and offer different QoS. These services were previously registered into the service registry (UDDI), and executed several times to populate the historical data base and enable the estimation of their QoS attributes.

Based on the analysis of the behaviour of Web services found on the Internet, response time of the candidate services was modified by adding random delays generated with a log-normal distribution. The distribution and its input values were determined after executing 5 services 1,000 times, collect their response times and analyze the difference between each execution.

The travel planning service was executed 50 times to analyze the behaviour of the optimization approach and evaluate its overall benefit. The maximum difference between estimated/real response time and cost was established as 10%. The service was also executed performing a simple service selection without QoS analysis.

As weights are those that provide priorities to the QoS attributes at the time of performing a service selection, values for w_1 and w_2 (algorithm in Fig. 2) were set as 0.3 and 0.7, respectively.

C. Evaluation Results

Initial results show that the proposed approach provides a meaningful improvement on the global QoS over a simple service selection approach. Global QoS refers to the final values of the different QoS properties (response time and cost) of the composite service. Fig. 5 and Fig. 6 present a comparison between both approaches based on response time and cost, respectively.

The first plot shows that the measured response time of the composite service executed using the optimization approach is closer to the corresponding estimated values, as compared to the behaviour of the simple selection approach, where most of the values are above the estimations. Measured average response time values correspond to 7049 ms and 7416 ms, where the proposed approach provides a mean reduction of 5%, a highest reduction of 14% and standard deviation of 7.45%.

Contrary to the behaviour of response time, cost estimations for the proposed approach are not close to the real measurements. As illustrated in Fig. 6, most values are above estimations; nevertheless, there can be found some significant cost reductions, the highest being of 16%. Average cost value was 452, with a standard deviation of 6.8%.

 ISBN: 978-1-61208-219-6

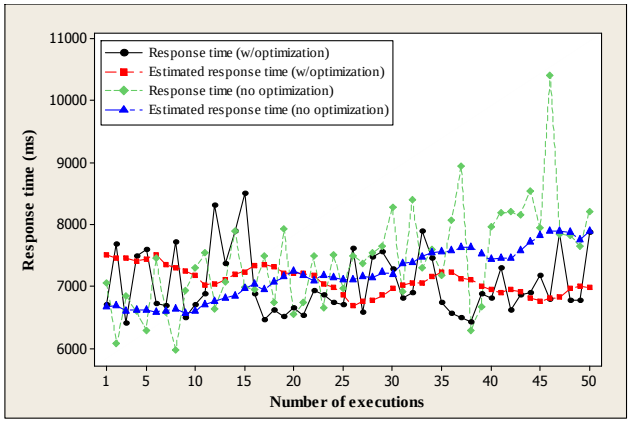


Figure 5. Composite service response time comparison between optimization and simple selection approaches.

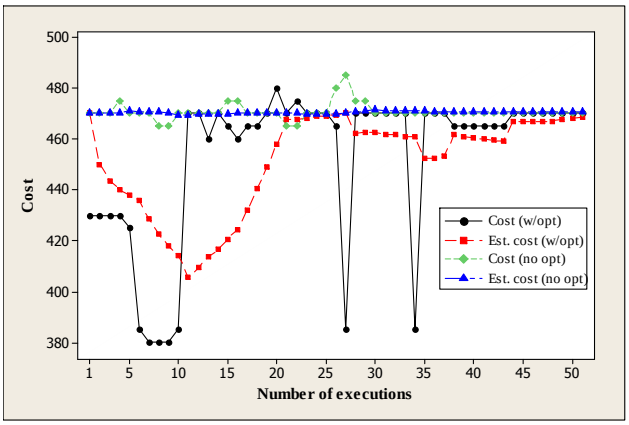


Figure 6. Composite service cost comparison between optimization and simple selection approaches.

To summarize the behaviour of both approaches, Fig. 7 presents a plot where response time and cost values were normalized and related using the Simple Additive Weighting formula presented in Section III. For both QoS attributes weights were established at 0.5.

From a global perspective, results demonstrate that using the proposed approach provides better QoS values, in most of the service executions.

It was noticed during the evaluation stage, that the overhead caused by the use of a service registry and predictive algorithms oscillate between 1500 and 2000 ms, which represent an important delay at runtime.

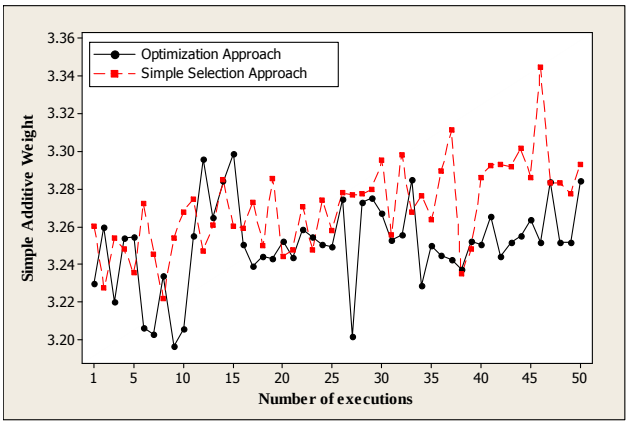


Figure 7. Composite service Simple Additive Weight comparison between optimization and simple selection approaches.

V. CONCLUSION AND FUTURE WORK

The execution of a composite service can be compromised by changes in the behaviour of its components. Mechanism such as adaptation, focus upon reducing the impact of these changes.

Adaptation in service composition aims to maintain/improve functional and quality levels while executing composite services. Thus, the development of adaptation mechanisms for service composition is an important task.

This work presents an adaptation approach for service composition that implements a self-optimization mechanism. During composite service execution, QoS attributes are monitored and optimization is triggered if there is a difference between estimated and real values.

In summary, evaluation indicates that by using the proposed approach, there can be achieved significant improvements in the global QoS of the composite services.

This paper is part of an ongoing research. Future work includes the extension of the quality criteria, considering other key QoS attributes like availability and reliability. Also, it is planned to investigate different self-adaptive properties and extend the actual framework, in order to increase the coverage of events that can occur at runtime.

REFERENCES

[1] W3C Working Group. "Web Services Architecture". Available: http://www.w3.org/TR/ws-arch/ [May, 2012].

[2] S. Dustdar and W. Schreiner, "A survey on web services composition," International Journal of Web and Grid Services, vol. 1, pp. 1–30, 2005.

[3] B. H. Cheng, et al., "Software Engineering for Self-Adaptive Systems: A Research Roadmap," Software Engineering for Self-Adaptive Systems, Lecture Notes In Computer Science, vol. 5525, pp. 1-26 2009.

[4] D. Ardagna and R. Mirandola, "Per-flow optimal service selection for Web services based processes," Journal of Systems and Software, vol. 83, pp. 1512-1523, 2010.

[5] W3C Working Group. "QoS for Web Services: Requirements and Possible Approaches". Available: http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/ [May, 2012].

[6] S.-Y. Hwang, et al., "A probabilistic approach to modeling and estimating the QoS of web-services-based workflows," Information Sciences, vol. 177, pp. 5484-5503, 2007.

 ISBN: 978-1-61208-219-6

[7] J. Cardoso, et al., "Quality of service for workflows and Web service processes," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, pp. 281-308, 2004.

[8] L. Zeng, et al., "QoS-Aware Middleware for Web Services Composition," IEEE Trans. Softw. Eng., vol. 30, pp. 311-327, 2004.

[9] P. Châtel, et al., "QoS-based Late-Binding of Service Invocations in Adaptive Business Processes," in Proceedings of the 2010 IEEE International Conference on Web Services, 2010, pp. 227-234.

[10] WS-Diamond Team, "WS-DIAMOND: Web Services-DiAgnosability, MONitoring and Diagnosis," MIT press, pp. 213-239, 2009.

[11] A. C. Huang and P. Steenkiste, "Building Self-Configuring Services Using Service-Specific Knowledge," in Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, 2004, pp. 45-54.

[12] D. Ardagna, et al., "PAWS: A Framework for Executing Adaptive Web-Service Processes," Software, IEEE, vol. 24, pp. 39-46, 2007.

[13] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," ACM Transactions on Autonomous and Adaptive Systems, vol. 4, pp. 1-42, 2009.

[14] Y. Dai, et al., "QoS-Driven Self-Healing Web Service Composition Based on Performance Prediction," Journal of Computer Science and Technology, vol. 24, pp. 250-261, March 2009.

[15] Y. Ying, et al., "A Self-healing composite Web service model," in Proceedings of the IEEE Asia-Pacific Services Computing Conference, 2009 (APSCC), 2009, pp. 307-312.

[16] V. Cardellini, et al., "MOSES: A Framework for QoS Driven Runtime Adaptation of Service-Oriented Systems," Software Engineering, IEEE Transactions on, vol. PP, 2011.

[17] D. Ardagna, et al., "A Service-Based Framework for Flexible Business Processes," Software, IEEE, vol. 28, pp. 61-67, 2011.

[18] D. Bianculli, et al., "Automated Dynamic Maintenance of Composite Services Based on Service Reputation," in Proceedings of the 5th international conference on Service-Oriented Computing (ICSOC '07), Vienna, Austria, 2007, pp. 449-455.

[19] L. Wenjuan, et al., "A framework to improve adaptability in web service composition," in 2nd International Conference on Computer Engineering and Technology (ICCET), Chengdu, 2010.

[20] A. Erradi and P. Maheshwari, "Dynamic Binding Framework for Adaptive Web Services," in Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services, 2008, pp. 162-167.

[21] G. Canfora, et al., "A framework for QoS-aware binding and re-binding of composite web services," The Journal of Systems and Software, vol. 81, pp. 1754-1769, 2008.

[22] R. Calinescu, et al., "Dynamic QoS Management and Optimization in Service-Based Systems," IEEE Trans. Softw. Eng., vol. 37, pp. 387-409, 2011.

[23] A. Erradi, et al., "WS-Policy based Monitoring of Composite Web Services," in Proceedings of the Fifth IEEE European Conference on Web Services, 2007, pp. 99-108.

[24] OASIS. "Web Services Business Process Execution Language Version 2.0". Available: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html [May, 2012].

 ISBN: 978-1-61208-219-6

Self-Adaptive Framework for Modular and Self-Reconfigurable Robotic Systems

Eugen Meister, Alexander Gutenkunst
Institute of Parallel and Distributed Systems
University of Stuttgart, Germany
Email: Eugen.Meister@ipvs.uni-stuttgart.de, Alexander.Gutenkunst@ipvs.uni-stuttgart.de

***Abstract*—In this paper, we introduce a framework for automatic generation of dynamic equations for modular self-reconfigurable robots. The equations for kinematics and dynamics are generated recursively in two steps by using geometrical formulations and recursive Newton-Euler method. This framework has the purpose to analyse the kinematics and dynamics for serial as well as for branched multibody robot topologies with different dyad structures. A multi-functional and easy to use graphical interface provides functionalities such assembling of topologies, visual feedback of trajectories and parameters editing. Two benchmark examples show, that the proposed framework results coincide with the results produced by classical Lagrangian method.**

***Keywords*-multi-body kinematics and dynamics; self-adaptive systems; automatic model generator.**

I. INTRODUCTION

Self-reconfigurable modular robots [1] open a spectrum of applications especially in dangerous and hazardous environments [2]. Self-reconfiguration is necessary when robots are operating completely autonomously without human intervention. Modular systems are on the one hand advantageous in comparison to specialized systems because they are adaptable to different situations and applications; however on the other hand, the complexity for modelling and control can grow rapidly.

In classical mechanics dynamical systems are usually described by setting up the equations of motion. The most common methods in the robotics are Newton-Euler, Lagrange, and Hamilton [3] formulations, all ending up with equivalent sets of equations. Different formulations may better suit for analysis, teaching purposes or efficient computation on robot.

Lagrange's equations, for example, rely on energy properties of mechanical systems considering the multibody system as a whole [4]. This method is often used for study of dynamics properties and analysis in control design.

More applicable on real robots are the Newton-Euler formulation of dynamics. In this method, the dynamic equations are written separately for each body. This formation consists of two parts describing linear (Newton) and angular (Euler) motion [5].

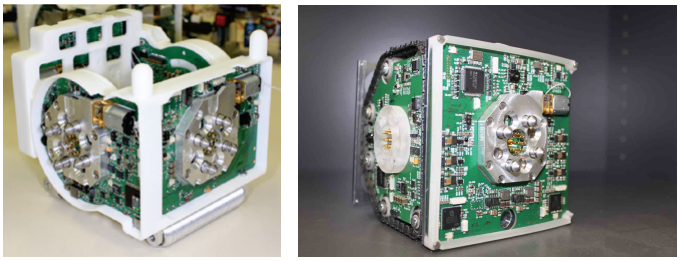
In case of modular reconfigurable multibody systems obtaining of equations of motions can be a challenging and time consuming task. In this paper we use a method using geometric formulation of the equation of motion originally introduced by Park and Bobrow [6]. This method is based on recursive formulation of robot dynamics using recursive Newton-Euler combined with mathematical calculus of Lie groups and Lie algebras. The description of motion is based on twist and wrenches summarizing angular and linear velocities as well as applied forces and moments in six-dimensional vectors.

In this approach, the Newton's second law ($F = ma$) and Euler's equations are applied in two recursions: the forward (outward) and the backward (inward) recursion. Therefore, we speak about two-step approach. In the forward recursion the velocities and accelerations of each link are iteratively propagated from a chosen base module to the end-links of multibody system. During the backward recursion the forces and moments are propagated vice versa from the end-link to the base forming the equations of motions step-by-step. Recursive derivation of the equations makes it applicable to different types of robot geometries and moreover allows automatizing the process. There exist several publications generalizing this method for variety of applications [7], [8], [9]. Most of efficient results use Newton-Euler algorithms, for example Luh, Walker, and Paul [10] expressing the equations of motion in local link reference frames and by doing this reduce the complexity from $O(n^3)$ to $O(n)$. This approach was lately improved by Walker and Orin [11] providing more efficient recursive algorithm. Featherstone [12] proposed the recursive Newton-Euler equations in terms of spatial notation by combining the linear and angular velocities and wrenches into six dimensional vectors (Plücker notation). His 'Articulated Body Inertia' (ABI) approach becomes widely accepted in current research and is also of complexity $O(n)$.

In the projects Symbrion [14] and Replicator [15], we develop autonomous modular reconfigurable robots that are capable to build multi-robot organisms by aggregating/disaggregating into different topologies [2]. In this paper we orientate our approach on the method proposed by Chen and Yang [16], which allows generating the motion equations in closed form based on Assembly Incidence Matrix (AIM) representation for serial as well as for tree-structured modular robot assemblies. The approach has been adapted to modular robots Backbone and Scout, because the geometry of modules differs from those proposed by Chen and Yang.

The paper is organized in the following way. In Section II, we give basic theoretical background about geometrical formulation for rigid body transformations. In Section III, we describe how the robot kinematics can be formulated for modular robots. In Section IV, robot assembly representation technique is introduced. Section V contains the recursive

 ISBN: 978-1-61208-219-6



(a) (b)

(c)

Fig. 1. Modular robots developed in projects Symbrion and Repicator [13]. (a) Backbone, (b) Scout, (c) Robots docked.

approach for calculation of dynamics equations. In order to evaluate the approach a graphical user interface (GUI) called MODUROB is built and is explained in Section VI. Finally, Sections VII concludes the work and gives a short outlook.

II. Theoretical Background

For kinematics analysis two Lie groups play an important role, the Special Euclidean Group $SE(3)$ and the Special Orthogonal Group $SO(3)$. $SE(3)$ group of rigid body motions consist of matrices of the form

$$\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}, \tag{1}$$

where $R \in SO(3)$ is the group of 3×3 rotation matrices and $p \in \mathbb{R}^{3\times 1}$ is a vector.

Lie algebra is also an important concept associated with the Lie groups. Lie algebra of $SE(3)$, denoted as $se(3)$, is a tangent space at the identity element of G. It can be shown that the Lie algebra of SE(3) consists of matrices of the form

$$\begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{2}$$

where

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{3}$$

Lie algebra is defined together with the bilinear map called Lie bracket, which satisfy following conditions:

- Skew-symmetry: $[a,b] = -[b,a]$.
- Jacobi identity: $[a,[b,c]] + [c,[a,b]] + [b,[c,a]] = 0$

If elements are square matrices, the Lie bracket is a matrix commutator $[A,B] = AB - BA$.

The connection between Lie Group $SE(3)$ and Lie algebra $se(3)$ is the exponential mapping, which maps $se(3)$ onto $SE(3)$. Exponential mapping allows an elegant way to formulate rigid body motions. The formula originates from the solution of the time-invariant linear differential equation for velocity $\dot{p}$ of a point that rotates about an axis ω

$$\dot{p}(t) = \omega \times p(t) = \hat{\omega} p(t). \tag{4}$$

By integrating the equation we receive

$$p(t) = e^{\hat{\omega} t} p(0), \tag{5}$$

where $p(0)$ is the initial position at $t = 0$ of the point. $\hat{\omega} \in so(3)$ is a skew symmetric matrix and the $e^{\hat{\omega} t}$ is the so-called the matrix exponential

$$e^{\hat{\omega} t} = I + \hat{\omega} t + \frac{(\hat{\omega} t)^2}{2!} + \frac{(\hat{\omega} t)^3}{3!} + \ldots \tag{6}$$

Considering rotations with unit velocity ($\|\omega\| = 1$), the net rotations can be formulated as follows:

$$e^{\hat{\omega} q} = I + q\hat{\omega} + \frac{q^2}{2!}\hat{\omega}^2 + \frac{q^3}{3!}\hat{\omega}^3 + \ldots \tag{7}$$

Using the Rodrigues' formula, a closed-form expression of this formula can be obtained without computing the full matrix exponent and therefore more efficient from the computational point of view.

$$e^{\hat{\omega} q} = I + \hat{\omega} + sinq + \hat{\omega}^2(1 - cosq). \tag{8}$$

The robot kinematics can be obtained by using the fact that rigid body motion can be achieved by a rotation about an axis combined with a translation parallel to it (Chasles's Theorem) [17].

In this case, the exponential mapping $e^{\hat{s} q}$ can be interpreted as an operator that transforms a rigid body from their initial pose to new pose combining rotations and translations at the same time

$$g_{ab}(q) = e^{\hat{s} q} g_{ab}(0), \tag{9}$$

where $g_{ab}(0) \in SE(3)$ is an initial pose and g_{ab} is the final pose. A twist associated with a screw motion is formulated as

$$s_i = \begin{bmatrix} -\omega_i \times p_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \tag{10}$$

where $\omega \in \mathbb{R}^{3\times 1}$ is a unit vector showing in the direction of the twist axis and $q_i \in \mathbb{R}^{3\times 1}$ is an arbitrary point on the axis. Revolute joints perform only pure rotations about an axis. Therefore the twist has the form:

$$s_i = \begin{bmatrix} 0 \\ \omega_i \end{bmatrix}. \tag{11}$$

Analogous, the pure translation is much simpler,

$$s_i = \begin{bmatrix} v_i \\ 0 \end{bmatrix}, \tag{12}$$

 ISBN: 978-1-61208-219-6

where $v_i \in \mathbb{R}^{3\times 1}$ is a unit vector facing in the direction of translation.

Linear mapping between an element of a Lie group and its Lie algebra can be performed by the adjoint representation. When X is given by $X = (R, p) \in SE(3)$, then the adjoint map $Ad_X : se(3) \mapsto se(3)$ acting on $y \in se(3)$ is defined by $Ad_X(y) = XyX^{-1}$. In [8] is also shown that $Ad_X(y)$ admits the 6×6 matrix representation

$$Ad_X(y) = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{13}$$

where $\hat{p}$ is the skew-symmetric matrix representation of $p \in \mathbb{R}^3$. Linear mapping between an element of Lie algebra and its Lie algebra can be performed via the Lie bracket

$$ad_x(y) = [x, y] \tag{14}$$

Given $x = (v_1, \omega_1) \in se(3)$, and $y = (v_2, \omega_2) \in se(3)$, the adjoint map admits corresponding 6×6 matrix representation

$$ad_x(y) = \begin{bmatrix} \hat{\omega}_1 & \hat{v}_1 \\ 0_{3\times 3} & \hat{\omega}_1 \end{bmatrix} \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix}. \tag{15}$$

Similar to twists that contain angular and linear velocities in one vector, wrenches or general forces are described in a similar way. Wrenches are vector pairs containing forces (angular components) and moments (rotational component) acting on a rigid body.

$$F = \begin{pmatrix} f \\ \tau \end{pmatrix}, \tag{16}$$

where $f \in \mathbb{R}^3$ is a linear force component and $\tau \in \mathbb{R}^3$ represents a rotational component. In contrast to general velocities as elements of $se(3)$, wrenches are acting on $se(3)^*$, the dual space and therefore behaves as covectors. For this reason wrenches transform differently under a change of coordinates by using so called adjoint transformation,

$$F_a = Ad^T_{g_{ba}} F_b, \tag{17}$$

where forces acting on the body coordinate frame B are written with respect to coordinate frame A. In spatial representation, this is equivalent as if the coordinate frame A were attached to the object.

III. Robot Kinematics

In modular reconfigurable systems the robot kinematics varies according to modules that are connected to each other. In homogeneous systems with the same physical parameters the kinematics depends only on the orientations of modules relative to each other. Such modular design is advantageous for autonomous systems. Using heterogeneous modules the complexity grows with the number of different modules that are used. Therefore, in most cases we assume identical or similar structure of the modules with similar physical properties. Both robots have been designed with similar geometry, same docking units and differ mostly in several insignificant properties such as number of sensors, different sensors or actuators. Nevertheless, even if the differences are not crucial, we speak about heterogeneous modules because of the additional Degree of Freedom (DOF) in Scout robot that is able to rotate the docking element even if only in limited way. Table I summarizes the mechanical properties of Backbone and Scout modular robots.

	Cubic Link Modules (Chen)	Backbone / Scout (Symbrion / Replicator)
Module types	homogeneous (large/small)	heterogeneous
Joint types	revolute, prismatic	revolute
# ports	6	4
DOFs	rot.: $\pm 180°$	Backbone: bend.:$\pm 90°$; Scout: bend.:$\pm 90°$, rot.: $\pm 180°$

TABLE I
Major differences between mechanical properties of Scout/Backbone robots.

Using only revolute joints without any prismatic joints simplify additionally the autonomous and recursive model generator for kinematics and finally for the dynamics model.

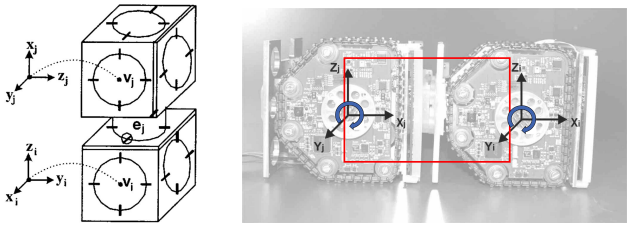
A. Dyad Kinematics

Dyad dependencies are common in recursive formulations because the calculation proceeds from one module to the next comprising only two modules. The calculation is done from the base module to all pendant links. In the approach proposed by Chen and Yang [16], a dyad is defined as two adjacent modules (v_i,v_j) connected by a joint e_j (Figure 2(a)). A link assembly is defined by taking one of those modules (link) together with one joint. The relative position and orientation of one frame attached to one module with respect to next frame in the second module can be described under joint displacement by a homogeneous 4×4 matrix $H_{i,j}(q) \in SE(3)$:

$$H_{i,j}(q_j) = H_{i,j}(0) e^{\hat{s}_j q_j}, \tag{18}$$

where $\hat{s}_j \in se(3)$ is the twist of joint e_j and q_j is the angle of rotation. The relative position and orientation between the modules can be recognized by the robot through different kind of on-board sensors such as accelerometers, compass or by vision system. In project Symbrion and Replicator the geometry of the Backbone (Figure 1(a)) and Scout (Figure 1(b)) robots differ from modules proposed by Chen and Yang. Backbone and Scout modules consist of two moving parts and one main hinge motor placed inside of each module and for this reason already implies a complete dyad as defined by Chen and Yang in each robot. In order to adapt the recursive kinematics approach to Backbone and Scout robot we need to extend the system boundaries of a dyad (Figure 2(b)). Since the most weight is concentrated in the middle of the modules where the main motors are placed, the attached coordinate frames for each module coincide with the centre of mass. Because of two robots and hence two revolute joints in a dyad only one joint is involved into calculation in each recursive step.

 ISBN: 978-1-61208-219-6



(a) (b)

Fig. 2. (a) A dyad defined by Chen and Yang [16], (b) Dyad for two Scout robots.

The orientation of axes of rotations depends on how robots are docked to each other. The relative pose can be described by 4×4 homogeneous matrix like in Eq. 18.

B. Forward Kinematics

Forward kinematics for modular reconfigurable robotic systems determines the poses of the end-links providing joint angles as an input. In this section, we introduce the modelling technique for forward kinematics based on local frame representation of the Product-of-Exponential (POE) formula originally proposed in [18] or in [8]. This technique can be easily applied to tree-structured robots with many branches (e.g., multi-legged robots). Based on recursive dyad kinematics, the calculation can be done simultaneously for all branches. In this paper, all robots are considered to be cube shaped robots based on Backbone or Scout geometries consisting of one major DOF. In general case, the forward kinematics for serial connected robots can be obtained for an arbitrary number of links by simply multiplying the exponential maps as follows:

$$g_{st}(q) = e^{\hat{s}_1 q_1} e^{\hat{s}_2 q_2} e^{\hat{s}_3 q_3} \dots e^{\hat{s}_n q_n} g_{st}(0), \tag{19}$$

where $\hat{s}_1$ to $\hat{s}_n$ have to be numbered sequentially starting with the chosen base module (Figure 3).

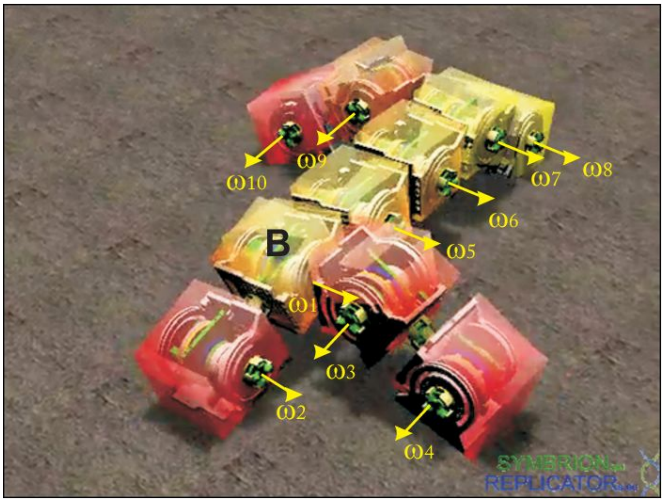


Fig. 3. Multi robot organism [19].

For a tree or branch structured robot configurations, the forward kinematics can be obtained in parallel way starting the calculation from a chosen base module to each pendant end-link in all branches. One possibility how the connecting order can be obtained is to use the AIM proposed by Chen and Yang [16]. For branched type of robots, two traversing algorithms are common to find the shortest paths: the Breadth-first search (BFS), and the Depth-first search (DFS) algorithms. The forward kinematic transformations for the branched robot configuration starting from base to each of the pendant links a_n of path k with m branches can be formulated as follows:

$$H(q_1, q_2, \dots, q_n) = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_k \\ \vdots \\ H_m \end{bmatrix} = \begin{bmatrix} \dots \\ \dots \\ \vdots \\ \prod_{i-1}^{n} (H_{a_{i-1}a_i}(0) e^{\hat{s}_{a_i} q_{a_i}}) \\ \vdots \\ \dots \end{bmatrix}, \tag{20}$$

where $H(q_1, q_2, \dots, q_n)$ represent all poses of all pendant end-links by using homogeneous 4×4 matrix representation.

IV. Robot Assembly Representations

Matrix notation is a powerful method to represent modular robots kinematic dependencies. The most common matrices used in robotics are the Adjacency and the Incidence matrix. Both matrices represent the connections between the neighbouring nodes. In [16], Chen proposes a method based on AIM that allows to represent the whole robot assembly consisting from links and joints additionally carrying the information about the type of robot and about used joints. A dynamic model for modular robot assembly is created autonomously from the AIM. This method was developed for a homogeneous kind of robots varying only in size with different joint possibilities including revolute or prismatic joints. Scout and Backbone robots contain only revolute joints however the number is not limited to one DOF. Therefore, the approach proposed in [16] cannot be directly used for this kind of modules and need to be adapted.

A. Adapted Assembly Incidence Matrix

The Backbone and the Scout robots are both cubic shaped robots, however provide only four sides that are equipped with docking units. Therefore, using the notation of gaming dice only ports $2-5$ are able to set a connection. A difference between modular robots proposed in [16] and the Scout/Backbone modules is that joints are not considered as a separate mechanical parts (joint modules), which are required to connect two modules, but rather are placed inside each of the modules. For these reasons each robot builds a full dyad already.

For simplicity, we allow docking only in horizontal plane and we also use the principle of gaming dice for side notations. Robot organisms have to go into initial configuration when additional robots decide to dock. Using this assumption, we distinguish between three major dyad configuration classes: the serial *DS*, the parallel *DP* and the orthogonal *DO* dyad

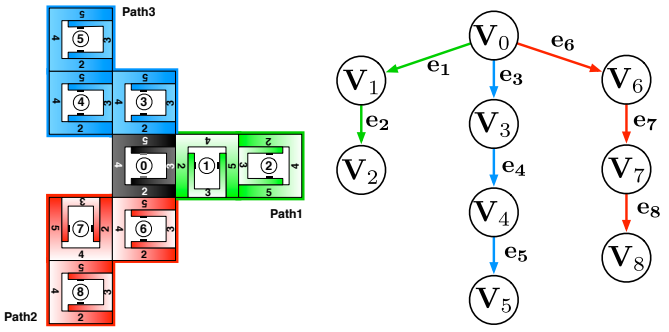
 ISBN: 978-1-61208-219-6

class, where the second letter determines the axes of rotation of module j with respect to module i. A serial coupled dyad (*DS*) is given when the axes of rotation are in one line. When the rotational axes are parallel than the dyad becomes a member of a parallel class (*DP*). Finally, when the axes are orthogonal to each other, the robots are classified as the orthogonal to each other connected robot assembly (*DO*). This information can be easily extracted from the matrix and used for direct computation. Additionally, the symmetry of the platform allows neglecting the sign of the orientation because it does not affect the calculation. Table II summarizes all possible configurations considering that top and bottom side of the robots and hence the sides 1 and 6 of a gaming dice do not contain docking units.

Set *DS*: Dyad: Serial Axes		**Set *DP*:** Dyad: Parallel Axes		**Set *DO*:** Dyad: Orthogonal Axes	
1st Mod.	2nd Mod.	1st Mod.	2nd Mod.	1st Mod.	2nd Mod.
2	2	3	3	2	3
2	5	3	4	2	4
5	2	4	3	3	2
5	5	4	4	3	5
				4	2
				4	5
				5	3
				5	4

TABLE II
DYADS LOOK-UP TABLE FOR SCOUT AND BACKBONE.

The autonomous docking procedure is based either on IR sensor communication or also can be fulfilled by using vision system [20], [21]. Backbone and the Scout robots have one revolute joint as a major actuator, therefore the information about the kind of actuators in the last row of the AIM is unnecessary. Instead, we use the last row for the three types of docking orientations for serial, parallel or orthogonal case. The last column in the AIM contains the information about the kind of robot, which is used. We denote the modified AIM as AIM_{SB}, where index '*SB*' denotes the first letters of both robots: the Scout and the Backbone robot.



(a) (b)

Fig. 4. (a) Multi robot organism example, (b) Directed graph representation.

In Figure 4, a small example of an organism and the corresponding graph is shown. The AIM_{SB} for this organism is shown in Figure 5.

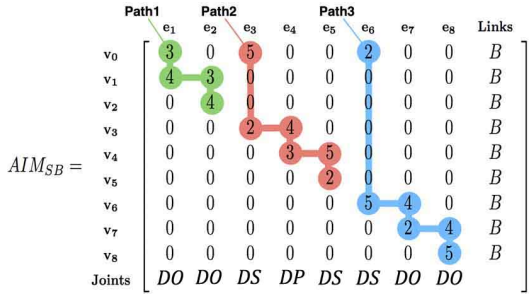


Fig. 5. AIM of robot assembly from Figure 4(a).

B. Direct/Indirect Recursive Transformations

Structuring the kinematics dependencies into an AIM_{SB}, we are able to apply the transformations T_{ij} between the modules directly once the AIM is determined. By reusing the already calculated dependencies that are stored into lists it is fast and efficient to calculate the kinematics for big robot organisms. We use two lists: one list containing transformation results between consecutive joints, we call it a Direct-Transformation-List (DTL) and another list called Indirect-Transformation-List (IDTL) for non-consecutive transformations between joints however still in the same kinematics path.

In DTL as shown in Table III, each line represents one direct transformation. The first two columns indicate the connected modules and the last two columns hold the information, which sides are connected. IDTL contains the indirect transformations, which are calculated by two successive transformations ($T_{ij} = T_{ix} \cdot T_{xj}$). The first two columns denote the desired transformation. Next four columns hold two multiplied transformations that are stored in DTL or in IDTL. Both tables refer to the example shown in Figure 4.

DTL			
T_{ij}		Sides	
i	j	from	to
0	1	3	2
1	2	5	3
0	3	5	2
3	4	4	3
4	5	5	2
0	6	2	5
6	7	4	2
7	8	4	5

IDTL					
T_{ij}		= T_{ix}	·	T_{xj}	
i	j	i	x	x	j
0	2	0	1	1	2
0	4	0	3	3	4
3	5	3	4	4	5
0	5	0	4	4	5
0	7	0	6	6	7
6	8	6	7	7	8
0	8	0	7	7	8

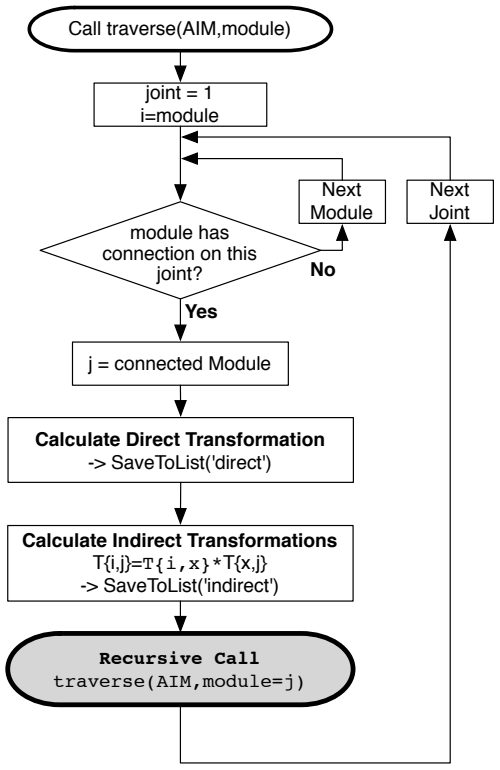
TABLE III
DIRECT AND INDIRECT TRANSFORMATION LISTS.

A short example demonstrates the first traversing calculations using both lists:

$$\begin{aligned} T_{01} &= T_{01}(0)e^{\hat{s}_1 q_1} && \text{direct} \\ T_{12} &= T_{12}(0)e^{\hat{s}_2 q_2} && \text{direct} \\ T_{02} &= T_{01} \cdot T_{12} && \text{indirect} \\ &\vdots && \vdots \end{aligned} \tag{21}$$

This algorithm can be compared with DFS algorithm, providing a flexible way to calculate the order in which all

possible transformations can be calculated during runtime.



(a)

Fig. 6. Traversing algorithm.

The flowchart of the algorithm is illustrated in Figure 6.

V. Modular Robot Dynamics

In general, two main branches of robot dynamics problems are mostly considered, namely the forward and the inverse dynamics problems. Forward dynamics play an important role in simulation of multibody systems, also called as direct dynamics. Forward dynamics problem determines accelerations and external reaction forces of the system giving initial values for positions, velocities and applied internal/external forces, whereas the inverse dynamics problem determines the applied forces required to produce a desired motion. The first problem that appears in modular self-reconfigurable robotics is that the model of the robot assembly cannot be known a priori. Therefore, the robot should be able to generate its own model autonomously without human intervention.

A. Recursive Two-Step Approach

The original idea for recursive formulation and computation of the closed form equation of motion was introduced by Park and Bobrow [6]. The idea was extended by Chen and Yang by introducing the AIM. Starting with the AIM, that contains the information about how robots are assembled, the formulation of equations of motion is done in two steps: first applying forward transformation from base to the end-link, followed by the second recursion backwards from the end-link to the base module. Finally, we get the equation of motion in a closed-form. Before starting the recursion, some assumption and initializations should be done. In the first step, the system has to choose the starting module denoted as the *base* module. Starting from this module, the AIM is filled based on path search algorithms such as BFS or DFS. After the AIM is built and all paths are determined the recursive approach can be started.

- **Initialization:** Given V_0, $\dot{V}_0$, F^e_{n+1}

$$V_b = V_0 = (0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0)^T \tag{22}$$
$$\dot{V}_b = \dot{V}_0 = (0 \;\; 0 \;\; g \;\; 0 \;\; 0 \;\; 0)^T \tag{23}$$

- **Forward recursion: for** $i = 1$ **to** n **do**

$$H_{i-1,i} = H_i e^{\hat{s}_i q_i} \tag{24}$$
$$V_i = Ad_{H^{-1}_{i-1,i}}(V_{i-1}) + S_i \dot{q}_i \tag{25}$$
$$\dot{V}_i = Ad_{H^{-1}_{i-1,i}}(\dot{V}_{i-1}) - ad_{Ad_{H^{-1}_{i-1,i}}(V_i)} + S_i \ddot{q}_i \tag{26}$$

where V_b and V_0 denote generalized velocities expressed in the starting frame 0 and all other quantities are expressed in link frame i. F_{n+1} is the force acting on the end-link of chained robots. This values can either be estimated or read from force sensors attached to the robots.

- **Backward recursion:** for $i = n$ to 1 do

$$F_i = Ad^*_{H^{-1}_{i,i+1}}(F_{i+1}) - F^e_i + M_i \dot{V}_i - ad^*_{V_i}(M_i V_i) \tag{27}$$
$$\tau_i = s_i^T F_i \tag{28}$$

Here, M_i is the generalized mass matrix of the form

$$M_i = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & mI_3 \end{bmatrix}, \tag{29}$$

where $\mathbf{I}$ is 3×3 inertia matrix and I is the identity matrix. The non-diagonal terms are zero because in our case the center of mass coincides with the origin. F_i is the total generalized force traversed from link $i-1$ to i consisting of internal and external wrenches and τ_i is the applied torque by the corresponding actuator.

B. Equations of Motion

By expanding the recursive equations (25) to (28) in body coordinates, it can be shown that the equations for generalized velocities, generalized accelerations and forces can be obtained in matrix form:

$$V = TS\dot{q} \tag{30}$$
$$\dot{V} = T_{H_0}\dot{V}_0 + TS\ddot{q} + Tad_{S\dot{q}}V \tag{31}$$
$$F = T^T F^e + T^T M\dot{V} + T^T ad^*_V MV \tag{32}$$
$$\tau = S^T F \tag{33}$$

where

$$\begin{aligned}
\dot{q} &= column[\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n] \in R^{n\times 1} \\
\ddot{q} &= column[\ddot{q}_1, \ddot{q}_2, \ldots, \ddot{q}_n] \in R^{n\times 1} \\
V &= column[V_1, V_2, \ldots, V_n] \in R^{6n\times 1} \\
\dot{V} &= column[\dot{V}_1, \dot{V}_2, \ldots, \dot{V}_n] \in R^{6n\times 1} \\
F &= column[F_1, F_2, \ldots, F_n] \in R^{6n\times 1} \\
F^e &= column[F_1^e, F_2^e, \ldots, F_n^e] \in R^{6n\times 1} \\
\tau &= column[\tau_1, \tau_2, \ldots, \tau_n] \in R^{n\times 1} \\
S &= diag[S_1, S_2, \ldots, S_n] \in R^{6n\times n} \\
M &= diag[M_1, M_2, \ldots, M_n] \in R^{6n\times 6n} \\
ad_{S\dot{q}} &= diag[-ad_{S_1\dot{q}_1}, -ad_{S_2\dot{q}_2}, \ldots, -ad_{S_n\dot{q}_n}] \in R^{6n\times 6n} \\
ad_V^* &= diag[-ad_{V_1}^*, -ad_{V_2}^*, \ldots, -ad_{V_n}^*] \in R^{6n\times 6n}
\end{aligned}$$

The index n represents the number of elements containing also virtual joints that are required to move the robot in a space [22].

$$T_{H_0} = \begin{bmatrix} Ad_{H_{0,1}^{-1}} \\ Ad_{H_{0,2}^{-1}} \\ \vdots \\ Ad_{H_{0,n}^{-1}} \end{bmatrix} \in R^{6n\times 6} \tag{34}$$

$$T = \begin{bmatrix} I_{6\times 6} & 0_{6\times 6} & 0_{6\times 6} & \cdots & 0_{6\times 6} \\ Ad_{H_{1,2}^{-1}} & I_{6\times 6} & 0_{6\times 6} & \cdots & 0_{6\times 6} \\ Ad_{H_{1,3}^{-1}} & Ad_{H_{2,3}^{-1}} & I_{6\times 6} & \cdots & 0_{6\times 6} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Ad_{H_{1,n}^{-1}} & Ad_{H_{2,n}^{-1}} & Ad_{H_{3,n}^{-1}} & \cdots & I_{6\times 6} \end{bmatrix} \in R^{6n\times 6n}, \tag{35}$$

where T is the transmission matrix for the whole robot assembly. The elements $H_{i,j}$ in T_{H_0} and in T can be read out directly from the DTL and IDTL lists.

The closed-form equation of motion of the classical form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q) = \tau \tag{36}$$

is obtained by substituting the equations 30 to 33, where $M(q)$ is the mass matrix; $C(q,\dot{q})$ describes the Coriolis and centrifugal accelerations and $N(q)$ represents the gravitational forces as well as the external forces.

$$M(q) = S^T T^T M T S \tag{37}$$

$$C(q,\dot{q}) = S^T T^T (M T ad_{S\dot{q}} + ad_V^*) T S \tag{38}$$

$$N(q) = S^T T^T M T_{H_0} \dot{V} + S^T T^T F^e \tag{39}$$

VI. MODUROB - Modular Robotics Software Tool

MODUROB is a tool built in MATLAB® that contains a possibility to build robot topologies by simply clicking on Topology Matrix Grid (Figure 8). Currently, two types of robots are provided: the Backbone (Figure 1(a)) and the Scout robot (Figure 1(b)). For simplification, robots are only allowed to assemble or disassemble in planar configurations on the ground. The automatic model can be built in two ways: analytically or numerically. The symbolic formulation in MATLAB is done by using the symbolic toolbox. For solving of differential equations the user can choose between the numerical integrators that are provided by MATLAB. In order to move the robot in a joint space, different gait generators are provided either using rhythmic generators based on rhythmic functions [22] or gait generators that use chaotic map. We use an approach proposed in, [23], that allows to generate periodic gaits that result from synchronization effects of coupled maps. Such approach can help to control complex multibody structures by mapping the active joints to an individual chaotic driver [24].

For evaluation or benchmarking of the framework two examples are implemented based on Lagrangian equations and can be compared with the geometrical approach. One example is a double pendulum example (Figure 8(a)) for example derived in [25] and the second is an extended pendulum that is movable on a shaft like a crane. In the second example, we use a virtual joint that allows moving the crane along one axes (Figure 8(b)).

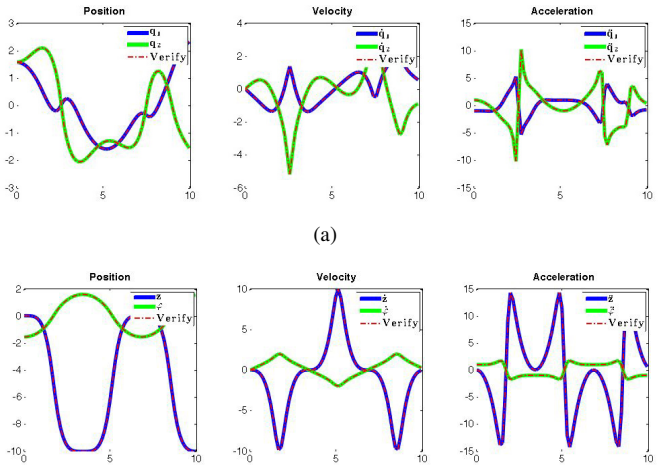
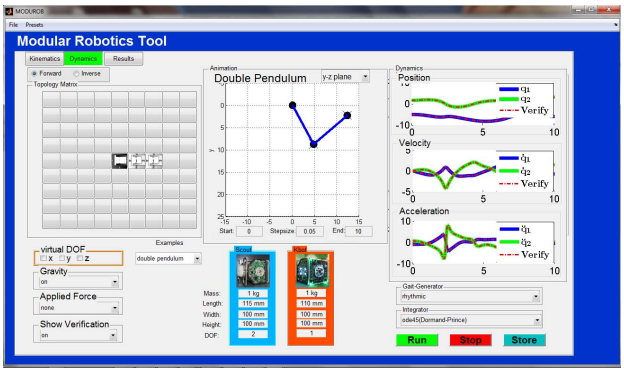


Fig. 7. Verification (dashed line) of geometrical POE approach with Lagrangian method using two examples: (a) Double pendulum model, (b) Crane model.

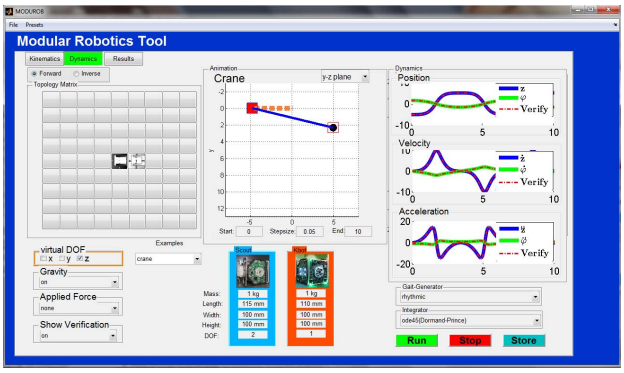
Both examples (Figure 7) show absolutely identical behaviour with examples implemented based on Lagrangian equations as well as with the geometrical approach based on twist and wrenches and therefore evaluates the approach.

VII. Conclusion and Future Work

In this paper, we demonstrate a MATLAB framework that allows analysing the kinematics and dynamics of modular robots. The calculation of self-adaptive models is based on recursive geometrical approach built on Screw Theory [26]. The proposed algorithm is inspired by the work from Chen and Yang and has been modified and adapted to the needs of robot



(a)



(b)

Fig. 8. (a) Implemented benchmark example of a double pendulum [25], (b) Crane example.

modules developed in projects Symbrion and Replicator. Such tool can not only be used for studying of topology behaviours of modular robots but also open a easy way to understand the theory behind the geometrical recursive approach. After we are able to build the models for kinematics and dynamics autonomously the next step will be to investigate different control design strategies such as feedback linearisation, self-organized and learning control mechanisms.

ACKNOWLEDGMENT

The "SYMBRION" project is funded by the European Commission within the work programme "Future and Emergent Technologies Proactive" under the grant agreement no. 216342. The "REPLICATOR" project is funded within the work programme "Cognitive Systems, Interaction, Robotics" under the grant agreement no. 216240.

REFERENCES

[1] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S. Chirikjian. Modular self-reconfigurable robot systems – challenges and opportunities for the future. *IEEE Robotics and Autonomation Magazine*, March:43–53, 2007.

[2] P. Levi and S. Kernbach, editors. *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer-Verlag, 2010.

[3] W. Greiner. *Classical Mechanics: Systems of Particles and Hamiltonian Dynamics*. Springer, 2010.

[4] R. A. Howland. *Intermediate dynamics: a linear algebraic approach*. Mechanical engineering series. Springer, 2006.

[5] M.D. Ardema. *Newton-Euler dynamics*. Springer, 2005.

[6] F. C. Park and J. E. Bobrow. A recursive algorithm for robot dynamics using lie groups. In *Proc. of IEEE Conference on Robotics and Automation*, pages 1535–1540, 1994.

[7] I.-M. Chen. *Theory and Applications of Modular Reconfigurable Robotics Systems*. PhD thesis, Calfornia Institute of Technology, CA, 1994.

[8] Li Z. Murray, R. M. and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.

[9] Scott Robert Ploen. *Geometric Algorithms for the Dynamics and Control of Multibody Systems*. PhD thesis, 1997.

[10] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul. On-line computational scheme for mechanical manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102(2):69–76, 1980.

[11] M. W. Walker and D. E. Orin. Efficient dynamic computer simulation of robotic mechanisms. *Journal of Dynamic Systems, Measurement, and Control*, 104(3):205–211, 1982.

[12] R. Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag, 2008.

[13] S. Kernbach, F. Schlachter, R. Humza, J. Liedke, S. Popesku, S. Russo, R. Matthias, C. Schwarzer, B. Girault, and ... P. Alschbach. Heterogeneity for increasing performance and reliability of self-reconfigurable multi-robot organisms. In *In Proc. IROS-11*, 2011.

[14] SYMBRION. *SYMBRION: Symbiotic Evolutionary Robot Organisms, 7th Framework Programme Project No FP7-ICT-2007.8.2*. European Communities, 2008-2012.

[15] REPLICATOR. *REPLICATOR: Robotic Evolutionary Self-Programming and Self-Assembling Organisms, 7th Framework Programme Project No FP7-ICT-2007.2.1*. European Communities, 2008-2012.

[16] I.-M. Chen and G. Yang. Automatic model generation for modular reconfigurable robot dynamics. *ASME Journal of Dynamic Systems, Measurement, and Control*, 120:346–352, 1999.

[17] J.M. Selig. *Geometric Fundamentals of Robotics*. Monographs in Computer Science. Springer, 2005.

[18] R. Brockett. *Mathematical theory of net-works and systems*, chapter Robotic manipulators and the product of exponential formula, pages 120–129. Springer, New York, 1984.

[19] Lutz Winkler and Heinz Wörn. Symbricator3D A Distributed Simulation Environment for Modular Robots. In Ming Xie, editor, *Intelligent Robotics and Applications, Lecture Notes in Computer Science*, pages 1266–1277, 2009.

[20] T. Krajník, J. Faigl, Vonásek V., Košnar K., M. Kulich, and L. Přeučil. Simple, yet Stable Bearing-only Navigation. *Journal of Field Robotics*, October 2010.

[21] Reza Saatchi M. Shuja Ahmed and Fabio Caparrelli. Support for robot docking and energy foraging - a computer vision approach. In *Proc. of 2nd International Conference on Pervasive and Embedded Computing and Communication Systems*, 2012.

[22] E. Meister, S. Stepanenko, and S. Kernbach. Adaptive locomotion of multibody snake-like robot. In *Proc. of Multibody Dynamics 2011, ECCOMAS Thematic Conference*, 2011.

[23] S. Kernbach, E. Meister, F. Schlachter, and O. Kernbach. Adaptation and self-adaptation of developmental multi-robot systems. *International Journal On Advances in Intelligent Systems*, 3:121–140, 2010.

[24] S. Kernbach, P. Levi, E. Meister, F. Schlachter, and O. Kernbach. Towards self-adaptation of robot organisms with a high developmental plasticity. In J. Guerrero, editor, *Proc. of the First IEEE International Conference on Adaptive and Self-adaptive Systems and Applications (IEEE ADAPTIVE 2009)*, pages 180–187, Athens/Glyfada, Greece, 2009. IEEE Computer Society Press.

[25] Javier E. Hasbun. *Classical Mechanics With MATLAB Applications*. Jones & Bartlett Publishers, 1 edition, March 2008.

[26] R. Ball. *A Treatise on the Theory of Screws*. Cambridge University Press, 1900.

 ISBN: 978-1-61208-219-6

EC4MAS: A Multi-Agent Model With Endogenous Control for Combinatorial Optimization Problem Solving

Gaël Clair, Frédéric Armetta and Salima Hassas
GAMA Laboratory,
Université Lyon 1
Lyon, FRANCE
Email: (gael.clair; frederic.armetta; salima.hassas) @univ-lyon1.fr

***Abstract*—Since a couple of years, new approaches are proposed to solve combinatorial optimization problems: multi-agent systems. In this paper, we propose a new model, EC4MAS, to build self-organizing multi-agent systems with more endogenous control. We start presenting a representative set of solving methods and we highlight what are the key elements of these solving processes and how they are used to construct a new representation of the problem to solve it. Generally, this representation is based on the characteristics of the method implemented but the construction of this representation could happen in the system without so much external intervention. This has been illustrated by some work in psychology that we present. Based on this observation, we propose and illustrate in this work a self-organizing multi-agent approach that tries to construct itself this representation, in an endogenous way. It is organized into a social organization of the different local solving behaviors and a spatial organization that represents the different structural/topological characteristics of the problem. The objective of the system is thus to find a good coupling between these two organizations to get the best possible representation.**

Keywords-multi-agent system; self-organization; endogenous control.

I. INTRODUCTION

In this paper, we present a model for more endogenous control in multi-agent systems for combinatorial optimization problem solving. First, we will present standard methods to solve this kind of problems. We highlight how they try to construct a new representation of the problem. Then we define how to create such representation in a multi-agent system to control it. We propose to set up an endogenous control system in order to design self-organizing systems not limited to a single kind of problem. Such control allows the system to adapt its own behavior according to its environment. This is similar to the natural process of learning and cognitive development which allow an individual to create his own knowledge to adapt more easily to problems. Our model introduces this development in multi-agent systems through the use of social representations.

We developed our model on this basis and we expose in this paper how we define all the elements to get a suitable representation for the control. Our model to create Endogenous Control for self-organized Multi-Agent System, EC4MAS, is based on a social and a spatial organization of agents and their coupling. These two organizations provide informations about the current solving strategy and on its outcome while the coupling allows the control system to dynamically adapt its own strategy.

Section II is a description of standard methods for optimization problem solving. Section III explains the notion of representation in these methods and informatics and psychology. The proposed model is then defined in Section IV. Section V presents and discusses results of experiments on a graph coloring problem. Section VI concludes the presented work and draws some perspectives.

II. SOLVING COMBINATORIAL OPTIMIZATION PROBLEMS

Combinatorial optimization is a domain which main interest is the solving of complex problems with high combinatory structure. We point out in this section well-known solving methods for this kind of problem.

Constructive approaches are initialized with a partial solution, generally empty, and try to build a complete solution widening the partial solution one variable at once. In these methods the only possible mean to direct the search is often the next variable to assign and its value. Like in greedy or backtracking methods some heuristics could help to determine the next variable to consider. Branch and bound method [1] is an implicit enumeration of the solution space, that is all the possible solutions can be examined but thanks to pruning techniques it can avoid to explore large subsets of bad solutions. These methods use local information and do not consider the global optimality making these more approximative ones.

Methods using the concept of neighborhood start with a complete assignment, which is not necessarily a solution, and make some changes to reach a different configuration. Changes needed to obtain new configurations define the possible neighbors, it is called the neighborhood function. Local search or Tabu search [2] are basic ones. In these methods one variable is changed at one time and we could use mechanisms like a tabu list to forbid examination of previous variables in a fixed period of time to get out from local optimum more efficiently. In Simulated annealing [3], neighbors are generated, evaluated and selected or not. Acceptation of a neighbor is conditioned by its improvement level and the moment of the search. A

 ISBN: 978-1-61208-219-6

temperature is decreased and used to get acceptation level, it is the mean to control and direct the evolution of the search. In these methods locality and local optimum are also problematic and mechanisms such as tabu list have to be used to improve the quality of solutions found. Simulated annealing tries to deal with an important problem, the balance between exploration and exploitation.

Evolutive algorithms are based on individual natural evolution principle. They are based on a population which is a set of individuals where each one represents a possible solution, an evaluation function which measures the adaptation level of one individual to its environment and an evolution process with some operators. An initial population is randomly generated, then each individual is evaluated and some of them are selected, finally new individuals are created using the evolution process. Algorithms using evolution strategies had been initially proposed by [4]. In Genetic algorithms [5] evolution operators, such as crossover and mutation are applied at random on one or several selected individuals. Genetic programming [6] uses a coding no more generic but specific to the handled problem, so is the only operator used, mutation. Here, population is the basis for the solving like in multi-agent systems but the use of individuals in these two approaches are different. Evolutive approaches make the population evolve in a centralized manner and select some individuals to survive which is not always the case in the multi-agent systems where agents are autonomous.

Besides standard solving methods, there are many other methods which want to use complex systems characteristics, like distribution of the solving process, and they need to be considered from this point of view. We can cite Ant colony optimization algorithms [7] and Particle swarm optimization [8]. Ant colony optimization algorithms are based on the ant natural behaviors where ants could solve a problem (finding the shortest path) indirectly communicating with only pheromones, it is called stigmergy. In Particle swarm optimization the first objective is to represent social interactions between agents which have a given objective in a common environment. It is important to notice that these methods are not evolutionary ones at the literal sense because they use cooperation between individuals instead of competition and finally no selection is done on the population. They are quite similar to multi-agent systems from this point of view.

III. Representation and control

In this section, we explicit the global principle used to solve a problem which is to construct a new representation of it.

A. Construction of a representation for the solving process

When we want to build a solving method, we have to construct a new representation of the problem to work with. This new representation is a mean to understand the problem and to define all the elements we want to use to get a solution.

First, the search space and its characteristics (roughness, dynamic, wideness ...) is used as a support for the solving process since it defines the elements to consider.

Then, the neighborhood function defines how the solving process gets from a solution to another, and is directly dependent on the search space characteristics.

Finally, the evaluation function, based on the nature or the type of expected objective, optimal or not for example.

B. Psychology and control

Creating a representation is a way to better understand a problem in order to solve it with limited capabilities. This principle is the one used by real individuals to solve real problems in their life. In this case the representation can be seen as the intelligence of this individual.

Cognitive development has been studied by Piaget [9]. He said that intelligence is no more than a more elaborate form of biological adaptation of an individual to its environment. It is a continuous process that rebalances structures of intelligence (schemas and operations) using two parallel processes, assimilation to interpret new facts and accommodation to change the cognitive structure.

Jean-Claude Abric [10] defines social representation "*as a functional vision of the world, which allows the individual or group to make sense of his actions, and understand the reality through its own reference system, so to adapt to it, to find its place in it*". It can be interpreted as a decision-making tool as "*It becomes the framework by which the rest of attitudes and judgments are adjusted so that everyone is on the same line as the group*" [11] so it is an attractor from individuals' point of view and it controls them. For more formal details we refer the reader to [12].

C. How to use social representation for endogenous control

Control in a self-organizing multi-agent system guides agents' choices. This guidance can not be exclusively based on purely local information but on a wider vision because it is involved in a group of agents which purpose can be seen as creating and maintaining a social representation.

Looking at the characteristics of social representations and those of an endogenous control, we see that these two concepts have much in common. Both systems have the task to control individuals' decisions in the group, dynamic and emerging characteristics. On this basis of strong ties we propose to use them to build a new model for endogenous control for multi-agent systems.

IV. Endogenous control for multi-agent system

In this section we present the construction of our model for endogenous control in self-organizing multi-agent based complex problem solvers (EC4MAS), its components, the social organization, the spatial organization and the coupling, and how they interact.

A. General organization of the model

Like seen in the previous section, to solve a problem we have to build a new representation of it, so the solving process can understand it to find a solution. The endogenous characteristics of our control means that a minimum, or in

 ISBN: 978-1-61208-219-6

the best case no, external interventions are needed to build the control system (or representation). In order to define the general organization of our model, or the general organization of the representation of the problem, we use the problem decomposition of Fig. 1.

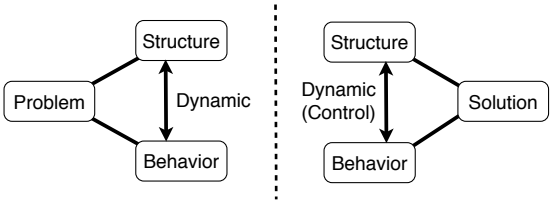


Fig. 1: Problem and solution organization

A problem is based on three key elements. First we can get informations on its structure which regroups directly available informations, like variables, domain of the variables, direct constraints and so on. The behavior mostly regroups indirect informations like influences between variables, indirect constraints or search space structure. These characteristics of the problem define how the problem will behave when we use its structure to solve it. Finally, the dynamic of the problem is the link between its structure and its behavior, it appears with the solving and it is the mechanism that links the elements of the structure to the ones of the behavior.

We used this problem organization to define the global organization of our model. The representation structure of the solution is used to model topological/structural characteristics of the current solution. The representation of the behavior of the solution is used to model the solving strategy currently used by the solving system. The dynamic which can be seen as the control of the solution is used to couple the structure and the behavior of the solution, to permanently adapt the current strategy to the current solution.

In a multi-agent system, all the agents only have access to limited informations and the global solution/strategy emerges from all the local acts or interactions of agents, so we model the structure and the behavior with organizations of agents. An organization can model an agent situation/strategy, or role, in the context of a particular situation/strategy and can use these relations to mark the mutual influences between them.

B. Structure: spatial organization

The structure or spatial organization is used to model the current situation in the search space. To get spatial informations on the problem structure we have to define several sensors. These sensors are used by the agents to perceive their spatial environment, so to define their spatial role. Agents can also communicate informations of these sensors to their neighbors. The spatial role reflects the current position of the agent in the search space and allows it to apprehend the difficulty of its situation. The spatial organization which can be observed and interpreted is based on:

- a set of m spatial roles $Rsp = \{Rsp_1, ..., Rsp_m\}$
- a configuration $Csp = \{a_1, ..., a_j\}$ with $Csp \in SP$ where a_i is an agent state and SP is the search space
- a function $fRsp : SP \rightarrow Rsp$

The organization of a group of agents in the environment or physical organizations of agents Csp, can highlight basic characteristics of the problem relevant to the solving. In order to capitalize these informations, it is necessary to allow their identification and use by the system. The spatial organization is based on the $fRsp$ function which associates a spatial role to an agent from the current spatial configuration. This function uses sensors, given to the agents to capture their situation, to determine the Rsp. The sensors could be specific to the problem to solve or more generic.

Spatial roles are about the effect of the solving process in the physical environment. The role is essentially descriptive of the problem and the situation of the system during its solving. The spatial organization connects particular configurations of the problem. In some cases, the problem definition may give access to specified elements to define spatial roles such as topology of the graph for graph coloring. In other cases this information is not identifiable from the outset but may appear during the search.

C. Behavior: social organization

The multi-agent system with multiple interacting agents, must be addressed in a more extended view than of a single agent. The overall activity is dependent on all individual actions but also on the interactions between agents. The group of agents is a reflection at a given time of the search activity of the system. This activity has to be captured by the system and has to be used to direct and control the research. Observation of these perceptions is defined by:

- $RSo = \{Rso_1, ..., Rso_n\}$, a set of n social roles
- $Lso = \{Lso_{11}, ..., Lso_{nn}\}$ where $Lso_{ij} = (Rso_i, Rso_j)$, $\forall i, j \in [1, ..., n]$, a set of relations between social roles
- $\forall Lso_i \in Lso, Cso = \{Lso_1, ..., Lso_i\}$, a set of social contexts
- a social organization $Oso =< Rso, Lso >$

Roles Rso act as guides for the agent to determine the appropriate strategy and dictate it a predefined behavior. The adoption of a social role by an agent implies that it adopts the guidelines and directives of that role. The action of the agent, so its social role choice, could have been influenced by other agents, this result in the relation Lso_{ij}. A relation Lso_{ij} exists if an agent with the role Rso_i has encountered another agent with the role Rso_j. The social organization involves a set of roles Rso and relations Lso between them. The activity of a single agent can not be isolated from other agents and therefore the social roles are linked within the organization. The social organization Oso gives information on situation of the agents within the solving process at a given time. The situation of an agent, and more precisely the adequacy of its social role, is directly dependent on its environment. To locate an agent of a social perspective, relations between it and its neighbors define a context Cso. The context is a part of the social organization, with a limited size around one particular agent. It provides information on relations between different social roles So in a particular situation of the search.

 ISBN: 978-1-61208-219-6

Formally, the social organization is used to represent the solving strategy of the system. Relations of social roles are based on the activity of the system, and more particularly, on the action of the agents. This dynamic updates permanently the organization to adapt the search.

D. Dynamic: coupling

Social and spatial organizations both provide information of a different nature. The first one is particularly interested in the mechanisms of the solving looking to the fittest behavior of the agents. The second one provides information on the status of the system in search space. These two elements are strongly linked because social roles define how agents act in the environment and spatial roles represent their situation in the environment. The coupling of these two organizations is defined by:

- a coupling function $fC : Cso \times Rsp \rightarrow \mathbb{R}$ with $\forall x \in Cso$ and $\forall y \in Rsp$, $0 \leq fC(x, y) \leq 1$
- a fitness function $fT : (Cso \times Rsp) \times Time \rightarrow \mathbb{R}$ where $Time$ is the number of cycles of the solving

The coupling fC is dynamic and allows the relations between spatial and social roles to evolve according to the fitness function fT evolution. To find the best couple *(Rso, Rsp)* for an agent in a given situation, is the key to success. This coupling is determined by evaluating and storing the pairs (social role, spatial role) created by agents during the search in the previous cycles (a cycle is an amount of time where each agent acts one time). A look back at previous choices with fT allows to update the coupling fC to adapt the control system.

E. EC4MAS principle

The general principle of our model is, to first represent the current situation (spatial organization), then to represent the current strategy (social organization) and finally, to couple these two organizations to permanently and dynamically adapt the solving process.

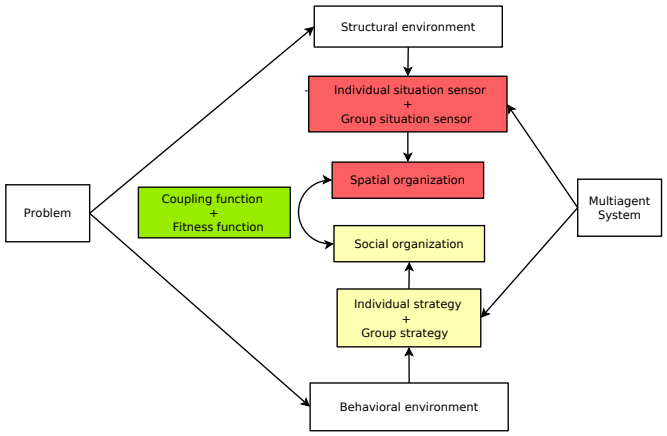


Fig. 2: EC4MAS Meta-Model

V. Results and Discussion

In this section, we present an example of the use of two different implementations of EC4MAS to solve a graph coloring problem.

A. Simplified version of EC4MAS

This version of the model is qualified as simplified because some shortcuts had been made as the roles and the organizations are predefined and explicitly implemented in the system.

1) Experimental Setup: Agents of the system represent the nodes of the graph to color. A solution is found when all the agents have no conflicts with their neighbors, so each two connected nodes are assigned different colors.

The main solving strategy of the agents is based on the Min Conflict heuristic [13]. Two social roles are used, the first is the exploitation strategy and the second is the exploration strategy. Exploitation tends to decrease the number of conflicts between an agent and its neighbors. Exploration can randomly take a color or apply the exploitation strategy (Min Conflict with exploration). The social organization is modeled with a static tree, where one role is represented at a level. The children are based on the representation amount percentage of the role in a situation divided in two (less or more 50%).

Spatial roles are based on the degree of the nodes (static). Each role regroups nodes with similar degree. The spatial organization follows the graph topology.

Coupling is done using matrices to link social and spatial organization. Matrices could be found as leafs in the social organization tree. These matrices have one column per social role and one row per spatial role. The values in the matrix are float numbers between 0 and 1 and are normalized on the row. An agent finds from its spatial role and its social context the associated matrix to get a social role. Higher the value in a social column, higher the probability for the social role to be selected is.

2) Results and analysis: To test our model we generated 100 different graphs with different seed. We used equi-partite 4-colorable graphs with 300 nodes, that means that the 4 color sets have the same size or can only be different from one node. An edge connectivity (ec) of 0,02333 (7/300) is used to get hard problems like seen in [14]. Each solving is done 1000 times with a maximum of 1000 cycles. The performance is the number of cycles to get a solution, if no solution is found 1001 is used.

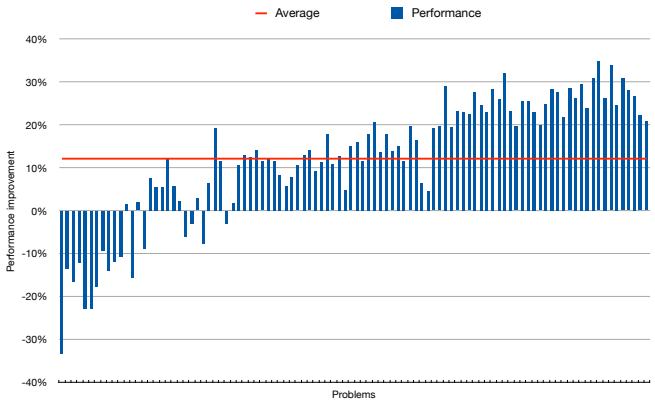


Fig. 3: Performance improvement of EC4MAS on 100 problems with 300 nodes, *ec* = 0,02333, 4 colors.

a) Performance: First, we randomly picked a problem (*ref-problem*) among generated graphs, a genetic algorithm is

 ISBN: 978-1-61208-219-6

used to get coupling values and to find the best exploring rate for Min Conflict with exploration (17,7% for *ref-problem*). Then we used this tuning to solve all the generated problems. Fig. 3 shows the performance improvement on all the 100 problems which is the difference between Min Conflict with exploration results and EC4MAS results. The problems are ordered by the number of cycles of the solving process. We can see that EC4MAS can generally improve the solving on a set of problems with similar characteristics since the average improvement of EC4MAS is about 12,4%. There is a big difference of performance between easier problems (at the beginning) and more difficult ones (at the end). The number of cycles to solve *ref-problem* with EC4MAS is 234, most of the problems with a negative improvement are under this value. This is due to the coupling used for *ref-problem* which is a representation of the problem and gives specific informations on it. With problems much easier than that, the solving process is too specialized and instead of guiding the search process it introduces too much perturbations.

TABLE I: Performance and efficiency

Method	Perf.	Tuning time	Efficiency
Min Conflict (17,7%)	100%	4	-
Optimal Min Conflict	124,71%	333	1,50%
EC4MAS (17,7%)	112,14%	22	20,39%

b) Tuning: In addition to the performance gain, we also focus on the tuning time of the system. Table I presents the performance gain of three different tunings and the efficiency of each method. The Min Conflict with 17,7% of exploration is taken as a reference for the measures. The tuning time is the sum of the time to find the optimal exploring rate for each problem for Optimal Min Conflict, and is the time to find the optimal exploring rate and the coupling values for *ref-problem* for EC4MAS. In the first case the tuning time is dependent on the number of problems and their hardness, in the second case only on the hardness of *ref-problem*. We can see here that the performance is much higher with optimal exploring rate, about 2 times more than EC4MAS, but the tuning time (in minutes) is about 15 times higher. In the end, the global efficiency (performance gain divided by tuning time) of EC4MAS is almost 13 times higher than Optimal Min Conflict. This shows that EC4MAS can learn the characteristics of a particular problem and is able to use this knowledge to really well solve similar problems with a limited tuning cost.

c) Genericity: ECM4AS uses the degree of nodes to create spatial roles. EC4MAS has been developed to be as generic as possible. To illustrate the genericity we introduce here a new type of sensor for spatial role, the local clustering coefficient. It is a good indicator for graph coloring problem hardness as seen in [15]. This coefficient is based on triangles between neighbors of the node and the node itself.

Table II shows the performance on *ref-problem* with the Min Conflict with exploration, EC4MAS with degree and clustering coefficient for spatial role. We can see that the most specific sensor which is the clustering coefficient is more efficient than the others, about 17% than Min Conflict while the degrees are only 11,5% better than Min Conflict. EC4MAS is generic

TABLE II: Improvement of spatial roles over Min Conflict

Spatial role	Perf. (cycle)	Improvement
Degree	234	11,5%
Clustering coefficient	223	17,04%

so it could support several types of sensors for spatial roles, from more general one like degree to more specific one like clustering coefficient.

B. EC4MAS

We present a second version of EC4MAS with no explicit spatial/social roles and organizations.

1) Experimental setup: Organizations are seen here like graphs, oriented graph for spatial organization, weighted graph for social organization and weighted and oriented graph for coupling.

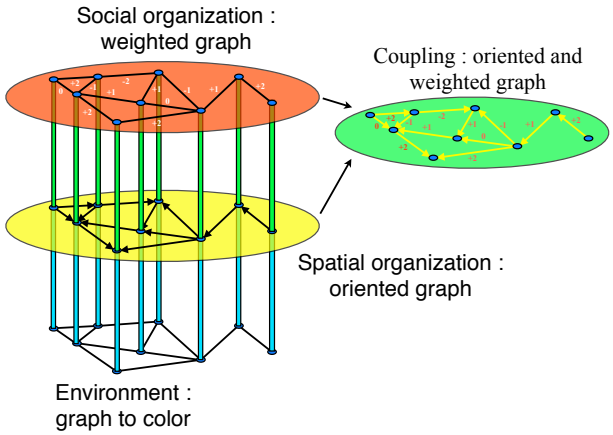


Fig. 4: Organizations of EC4MAS

In this version, spatial sensors are used to construct an oriented graph on the basis of the graph to color. They provide information on degree of the node and its color, so an agent can get a degree of freedom which can be interpreted as spatial role. When a color is selected by an agent, it may create some new conflicts. Social roles are interpreted as the global action of the agent when it chooses its color. If the color is not in conflict with the neighbors we can say that the conflicts are removed by the agent, if the number of conflicts of the agent is increased/decreased we can say that the agent add/remove some conflicts from the system. The impact of the choice of a color on the system is marked through the weights in the social organization (weight of the links between neighbors), increase of the weight if perturbations are removed and decrease of the weight if perturbations are created or transmitted. The coupling associates weights of the social organization to the oriented edges of the spatial organization, it represents the flow of the perturbations in the system. It may direct the perturbations through the best path (nodes and edges) to be sure that they will be removed as quickly as possible.

2) Results and analysis: We present here the first results of this version, further experimentations will be made in the

 ISBN: 978-1-61208-219-6

future. We compared this version to a model exposed in [16]. In this model agents compute in a centralized way, a value for their environment in a fixed range to get informations on the possibles conflicts introduced by a specific color, as in EC4MAS agents use the coupling graph.

TABLE III: Improvement of EC4MAS over Min Conflict

Model	Perf. (cycle)	Improvement
MinConflict	211	
multi-agent for K-coloring	180	15%
EC4MAS	201	5%

Table III presents results with Min Conflict, the model described in [16] and E4CMAS. We can see that the specific model for k-coloring improves the solving by 15% while our model only over 5%. We can explain this result by the high dynamic of the system during the solving. The evolution of the weights in the social organization has to evolve quickly to be always adapted to the current strategy otherwise, it slows down the solving process providing it no more appropriate data. A good setting of the different changes of the weights have to be found to get optimal informations for the system. The flow graph, or coupling, gives informations to decrease the number of perturbations (or conflicts) in the system but it has to be updated as quick as the system to be controled evolves, which is not totally the case in this version.

VI. Conclusion and future work

In this paper, we present our model, EC4MAS, for more endogenous control in multi-agent based solvers for combinatorial optimization problems. We started from well-know methods of solving for this kind of problems and used them to define a general approach to solve combinatorial optimization problems. The main idea is to define a new representation of the problem more easily understandable and more adapted to the limited knowledge we have at the moment we conceive the solving process. This representation is based on the problem structure, its behavior and its dynamic. If we want to build a new representation of the problem we have to take into account these elements and to base the conception process on them. Some works in psychology have shown that this process is the one used by an individual to develop his intellect. This emergence of a representation could also be seen in groups of individuals to understand and work with some concepts.

Our model, EC4MAS, is based on theses observations and uses it to construct a control system in an endogenous way. This model is based on a social and a spatial organization and their coupling. These two organizations provide information on the current solving strategy of the system and on its result, and the coupling allows the control system to dynamically adapt the strategy of the system more efficiently.

EC4MAS is a generic model and could be used to solve different kind of problems. The spatial organization could be adapted to use specific informations to let EC4MAS be able to solve different kind of problems. On more harder problems EC4MAS gives good improvements since the characteristics of the problem are used to better tackle it. EC4MAS makes the tuning of the system robust in front of changes in a problem and the coupling of social and spatial organization provides pertinent solutions to already encountered situations with a specific strategy. The tuning has not been changed when new problems are submitted. This is a great point because individual optimization is very expensive and could not be always used, more particularly when problems are dynamic. The endogenous self-organized characteristic of EC4MAS could efficiently limit the amount of time and resources to solve a problem, this is shown by the simplified version.

The second presented version shows us the importance of the speed of the adaptation of the coupling (or control) system. To get the system oriented the coupling must be as quick as possible to construct the representation, because it has to represent the current situation/strategy and also the consequences of the previous ones to guide the system.

In Future work, we will focus on this speed characteristic and we will implement the model with new problems like the jobshop problem.

References

[1] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica: Journal of the Econometric Society*, pp. 497–520, 1960.

[2] F. Glover and M. Laguna, *Tabu search*. Springer, 1997.

[3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[4] I. Rechenberg, "Cybernetic solution path of an experimental problem," Royal Air Force Establishment, Tech. Rep., 1965.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, 1992.

[6] N. Cramer, "A representation for the adaptive generation of simple sequential programs," in *Proceedings of the 1st International Conference on Genetic Algorithms table of contents*. L. Erlbaum Associates Inc. Hillsdale, NJ, USA, 1985, pp. 183–187.

[7] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the First European Conference on Artificial Life (ECAL)*, F. Varela and P. Bourgine, Eds. MIT Press, Cambridge, Massachusetts, 1991, pp. 134–142.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, 1995, pp. 1942–1948.

[9] J. Piaget, *Psychologie de l'intelligence*. Paris: Armand Colin., 1947.

[10] J.-C. Abric, *Pratiques sociales et représentations*. Presses Universitaires de France, 1994.

[11] S. Moscovici and W. Doise, *Dissensions et consensus: une théorie générale des décisions collectives*. Presses universitaires de France, 1992.

[12] L. L. Carvalho, G. Clair, S. Hassas, and E. M. Braga, *Essais (im) pertinents sur représentation mental et cognition*. Synopsis, 2011, ch. Représentations : Du Contrôle Endogène pour des Systèmes Complexes Adaptatifs.

[13] S. Minton, M. Johnston, A. Philips, and P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems," *Constraint-Based Reasoning*, vol. 58, no. 1-3, pp. 161–205, 1994.

[14] A. Eiben, J. Van Der Hauw, and J. van Hemert, "Graph coloring with adaptive evolutionary algorithms," *Journal of Heuristics*, vol. 4, no. 1, pp. 25–46, 1998.

[15] A. A. Tsonis, K. L. Swanson, and G. Wang, "Estimating the clustering coefficient in scale-free networks on lattices with local spatial correlation structure," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 21, pp. 5287 – 5294, 2008.

[16] B. Smati, F. Armetta, and S. Hassas, "Résolution par système multi-agents de problèmes combinatoires: une application à la k-coloration," Laboratoire LIESP, Tech. Rep., 2008.

 ISBN: 978-1-61208-219-6

Information Models for Managing Monitoring Adaptation Enforcement

Audrey Moui, Thierry Desprats, Emmanuel Lavinal, Michelle Sibilla
IRIT, Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse cedex 9, France
Email: {moui,desprats,lavinal,sibilla}@irit.fr

***Abstract*—Integrated management should cope with numerous, heterogeneous and complex systems in a multi-dimensional environment. In this context, the monitoring activity should be efficient and sensitive to variations of management applications requirements. In this paper, we define a framework which includes three capabilities that support monitoring adaptation. Particularly, we have defined information models for the first two capabilities, namely configurability and adaptability. This framework is modular enough to integrate any existing solution that proposes monitoring adaptation decisions. A partial CIM/WBEM implementation has been tested to measure the overhead due to the management cost of the proposed approach.**

***Keywords*—Monitoring; Adaptation; Information model; CIM/WBEM; Integrated management.**

I. INTRODUCTION

The multi-level aspect of integrated management (nodes, networks, systems, services), the dependence between heterogeneous components and the affluence of management information make the management activity more and more difficult to perform; indeed, new management paradigms are based on more and more autonomous and decentralized decision-making [14].

The efficiency of the monitoring activity has become a major preoccupation in various contexts such as integrated management of complex (networked) systems or autonomic and self-managed entities. In many cases, management paradigms are organized with the fundamental help of the classical MAPE loop (Monitor – Analyze – Plan – Execute) [7]. The managed system is observed (M) and an analysis (A) is performed to either detect or prevent failures or every relevant situations to be interpreted. In such a case, a reactive or proactive technical decision is taken and planned (P). Consequently, adjustment actions are then executed (E) on the system to hopefully improve its efficiency.

The quality of the control is provided by the quality of the implementation of each of these four functions, but also by the fullness of the interactions occurring between them. Particularly, at the first step of the cycle, the analysis function relies on the quality of the information (QoI) delivered by the monitoring mechanisms. One of the major issues is to improve this relationship by allowing a feedback from the analysis function to the monitoring one. This feedback constitutes a mean by which the analysis function, according to its management objectives, can continuously express requirements about the information and its quality, which are provided by the monitoring function. Dynamically taking into consideration the variations of these management functional requirements implies that the monitoring function has to be adaptive at runtime.

From an operational viewpoint, performing monitoring activities consumes several resources such as CPU, memory, energy, bandwidth, etc. This execution environment can also constrain the monitoring activities either directly, when resources are – partially or fully – temporally unavailable, or indirectly when resource consumption restrictions policies are applied (e.g., do not allow more than 10 % of global bandwidth for monitoring traffic). Consequently, the monitoring also needs to adapt itself to fit with any particular issues, which dynamically occur within the constrained operational environment.

Finally, a self-optimization concern may be supported by the monitoring function to increase the level of its efficiency. In this case, resulting from an introspective analysis of its performance, some self-adjusting decisions can be taken but enforced at the only condition that the monitoring function is adaptable.

This paper presents work performed to model the adaptation enforcement management, and a partial CIM/WBEM (Common Information Model [2] /Web-Based Enterprise Management [15]) implementation, which has been tested to measure the overhead due to the management cost of the proposed approach. It firstly presents our motivation and the global context of adaptation enforcement. The third and fourth sections present the modelling of the enforcement of monitoring strategy and the information models for adaptation enforcement management, respectively. The fifth section then presents the implemented prototype and the first measures, before concluding the paper.

II. MOTIVATION

For us, *adaptive monitoring* refers to the ability an online monitoring function has to decide and to enforce, without disruption, the adjustment of its behavior for maintaining its effectiveness, in respect of the variations of both functional requirements and operational constraints, and possibly for improving its efficiency according to self-optimization objectives.

 ISBN: 978-1-61208-219-6

A. Managing the Adaptation Enforcement

Automating monitoring adaptation requires to manage and to control the monitoring activity itself.

Independently of the deployment level, the monitoring activity is a process which relies on the gathering of raw, symbolic, aggregated, transformed or filtered data. Typically, gathering is based on the possibly combined use of "polling" (i.e., pulling data, by periodically requesting some targeted source or aggregator for data) and "event reporting" (i.e., receiving, in an ad hoc, sporadic or periodic way, pushed data from a provider or an aggregator) mechanisms.

Modifying the current behavior of a monitoring activity will finally result in achieving actions which will enforce adaptation decisions that aim at varying the scope and/or the modalities of the observations.

Reaching a CPU resource consumption reduction goal will generally lead to the decrease of the number of basic monitoring mechanisms which are running; at the opposite, trying to extend the scope of the monitoring will certainly cause new basic monitoring mechanisms to be launched (for example, giving the monitoring activity the possibility to question another managed element on the monitored underlying system gives an enlarged vision). Requiring a higher level of freshness of the collected data will cause the intensification of a polling frequency while limiting the monitoring traffic on the network will result in increasing the duration between two successive pull requests. For other purpose, like self-optimization, decision can be made to temporally suspend some particular mechanisms while considering some functional management needs (for example in a process of diagnostics refinement) will cause a temporal prolongation of a running mechanism. Illustrated by these previous typical situations, it can be concluded that, in the large majority of cases, the achievement of a monitoring adaptation will be materialized by managing the operational cycle of basic monitoring mechanisms and some of the parameters that can govern their behavior.

The main requirements to operate such a management of the enforcement of an adaptation decision include:

1) The need to capture any operational query about the adjustment of the behavior of the running monitoring activity. It implies to offer some well-defined interface allowing the decisional level to express actions to be done to the underlying enforcement infrastructure.
2) The necessity to verify at runtime the coherence and the feasibility of the requested actions. In particular, they can concern an updating of some settings governing the behavior of one or more running monitoring mechanisms. They can also affect the creation, the termination, or more generally, the operational state change of such a mechanism. Ensuring the consistency of the current state of the monitoring activity is another preoccupation. This must rely on the online availability of some information giving a management view of the state and the behavior of the monitoring mechanisms.
3) Finally, the ability to obtain some statistical data on each monitoring operations may serve for limitation resource consumption or self-optimization driven decisions making.

In this paper, we do not focus on the process of a monitoring adaptation decision making; we assume that it is based on any of well-known solutions including CSP, constraints resolution, inference engine, management policies, rules, bio-inspired approach, etc. In a complementary way, our concern is to focus more on the effective realization of the decided adaptation by providing a generic framework to support the management of this post-decision process.

B. Related Works

Numerous works, which are mainly dedicated to network monitoring, already contributed to enhance the monitoring adaptiveness level.

Duarte et al. [5] proposes a language for programming configurable monitoring applications, but which is dedicated to an SNMP/DISMAN environment; this facility is not generic enough to be used in every management context.

Massie et al. [9], Dilman et al. [3] and Moghé et al. [10] focus on finding protocol solutions and are mainly concerned with performance criteria (RAP [10] is only concerned with polling adaptability and Ganglia [9] provides the ability to reconfigure the monitoring mechanisms according to the newtork context, but only for small networks).

Some of the works have designed adaptive protocol-based solutions allowing trade-offs between performance and quality of the information [13], which exclusively relies on push data models. Other works target different area than newtork.

Xue et al. [16] deals with adaptive and scalable monitoring of cluster and propose facility for switching between a light and a heavy monitoring mode.

III. Modelling the Enforcement of Monitoring Strategy Adaptation

A. Preliminary definitions

- *Monitored Elements*
  Both polling and event reporting mechanisms are applied on system components.
  1) for polling mechanism, these components are called "targets" and can be seen as managed elements or sources of pulled data: these targets can be of different kinds:
     a) a whole object (e.g., a switch on a network) or a collection of objects (e.g., all the switches of a particular network);
     b) a property (e.g., the operational state of a(several) switch(es)) or a set of properties (e.g., any relevant information of a(several) switch(es));
  2) for event reporting, the components are considered as "sources" of pushed information or indications (e.g., warnings, alerts, particular data, pre-filtered or aggregated pushed information).

 ISBN: 978-1-61208-219-6

With concerns of genericity, we consider targets and sources as any relevant information related to managed elements. Let T be the set of these components within the scope of a monitoring activity, such as:

$$\forall n \in \mathbb{N}, \quad T = \langle t_1, t_2, ..., t_n \rangle$$

- *Basic Monitoring Mechanism*
  We can unify the polling and event reporting mechanisms (respectively named as P and E) as a set of basic monitoring mechanisms. This set is represented as M:
  $$M_T = P_T \cup E_T \neq \emptyset$$
- *Mechanism Configuration*
  Moreover, a basic monitoring mechanism must have to consider, initially and when changes are demanded on line, the values of a set of parameters which influence the way this mechanism operates in relation to its associated monitored elements. The set of these parameters will be represented as a configuration C, grouping two kinds of parameters: some are conceptually not modifiable, and the others can be modified.
- *Operational State*
  A mechanism owns an operational state Op which indicates if an active mechanism is currently $running$ or $pending$.

B. Monitoring Strategy

We define a *monitoring strategy* S as an association between the mechanisms M (polling or event reporting) applied on their respective targets, their configuration C and their operational state Op:

$$S = (M_T, C, Op) = \{\langle M_{t_1}, C_1, Op_1 \rangle, ..., \langle M_{t_n}, C_n, Op_n \rangle\}$$

C. Monitoring Strategy Adaptation

An adaptation will result in a monitoring strategy change: e.g., the modification of a temporal parameter value (for instance the polling periodicity value), the adjunction or the deletion of a target, the suspension or the deletion of the monitoring operation, etc.

A monitoring strategy change will consequently modify the monitoring activity behavior. The evolution from a strategy S to a strategy S' is defined as an *adaptation* δ_S [1].

$$\delta_S = S \rightarrow S'$$

An evolution of this strategy is then one of the following change (possibly combined) of the previously defined set (strategy formalization):

- The set of the mechanisms can be modified ($M \rightarrow M'$) by adding or deleting a monitoring mechanism M (polling or event reporting as well);
- The set of the targets can be modified ($T \rightarrow T'$) by adding or deleting a target T associated to a monitoring mechanism;
- The set of the configuration parameters can be modified ($C \rightarrow C'$) by updating the value of one or more configuration parameters C for a monitoring mechanism;
- The value of an operational parameter can be modified ($Op \rightarrow Op'$) by suspension or resumption of a monitoring mechanism.

These modifications have been explicited on Figure 3.

IV. Informational Models for Management of Adaptation Enforcement

A. Management Framework Overview

A particular environment has been defined to support the concepts of strategy and strategy adaptation. To do so, three capabilities have been determined as the requirements that a framework enforcing monitoring adaptation must ensure. These interconnected capabilities can be seen on Figure 1 and have been introduced in [12].

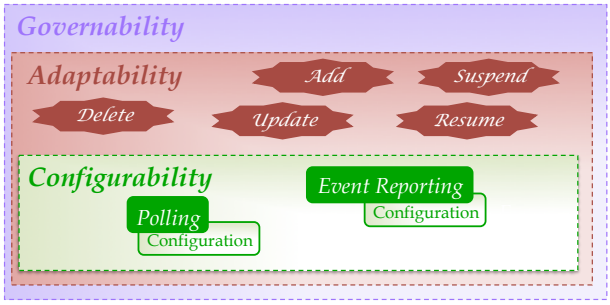


Fig. 1. Required Framework Capabilities

1) Configurability: The configurability is defined as the capacity of the monitoring mechanisms to be dynamically adapted: the configuration parameters governing the behavior of the monitoring mechanisms can be defined initially and then modified dynamically at runtime and without disruption when needed to obtain the best vision of the underlying system.

Defining a period parameter `PollingPeriod` for polling that we can modify at runtime is an example of configurability capability.

2) Adaptability: The adaptability is the ability to execute the monitoring adaptation. This capability is enforced by performing atomic adaptation operations over the underlying mechanisms configuration, with the objective to modify the current monitoring strategy. Consequently, this capability makes possible to dynamically modify at runtime the behavior of a monitoring activity.

An example of adaptability capability is to be able to dynamically modify the value of `PollingPeriod`.

3) Governability: Some intelligence is required to support the adaptation decision: the point is to decide if and how the monitoring activity has to be adjusted.

Piloting the adaptation operations to adjust and/or optimize the monitoring activity is the role of the governability capability. An adaptation decision can be taken in response to business objectives changes or for monitoring self-optimization, or in reaction to some constraints variations on technological resources availability.

This capability is an interface aiming at providing the inputs needed for the monitoring adaptation enforcement (the adaptation decisions). According to the previous example,

 ISBN: 978-1-61208-219-6

specifying the rule which will trigger the modification of the value of `PollingPeriod` is a concern of the governability level.

B. An Informational Model for Adaptability

We determined that a monitoring strategy is then a view at a particular moment of a set of monitoring mechanisms applied on their respective targets, their configuration parameters, their operational state, and optionally a set of statistical information data. Figure 2 sums up the concept of strategy.

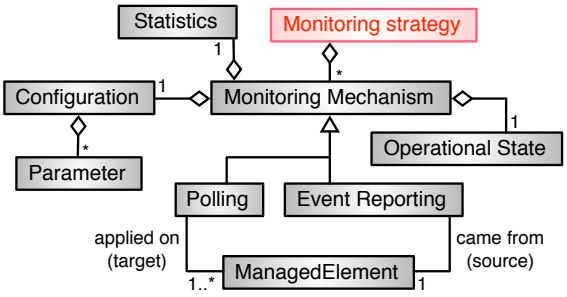


Fig. 2. Informational Model for Monitoring Strategy

Adaptability is needed to enforce the adaptation. Therefore, we have defined five basic operators, which may be identified as service interfaces which support at runtime the reconfiguration enforcement of the monitoring activities. These operators are atomic operations, which can also be combined as agregations.

1) *Adjunction*
   A mechanism, polling or event reporting, can be added by using the $\mathcal{A}$ operator: the configuration of the mechanism is required; the operational state will be automatically set to "active" (and the mechanism will be executed once created); and the managed element(s) on which the mechanism applies (target(s)) is(are) required.
2) *Deletion*
   A mechanism, polling or event reporting, can be deleted by using the $\mathcal{D}$ operator.
3) *Update*
   A configuration of a mechanism, polling or event reporting, can be updated at runtime, thanks to the $\mathcal{U}$ operator. This operator makes possible the modification of whole or part of its current configuration (all the modifiable parameters).
   Moreover, the list of the monitored elements can be modified: a target can be added to or removed from this list of targets at runtime and without disruption.
4) *Suspension*
   A polling mechanism can be used more than one time. Rather than deleting the mechanism, it can be relevant to suspend it, thanks to the $\mathcal{S}$ operator. Then the polling mechanism can be used again later if needed.
5) *Resumption*
   The resumption, performed with the $\mathcal{R}$ operator, corresponds to the reactivation of a previously suspended polling mechanism.

Figure 3 defines the inputs and outputs of these operators, in order to clarify their operational use.

Algorithms have been writen to take into account every possible constraints at runtime.

- Two levels of parameters have been determined. Selector parameters define the global behavior of the mechanism (e.g., the stop mode describes the way a polling ends); according to the value of this parameter, additional parameters need to be filled to complete the behavior description (e.g., when the stop mode is set to "Iterative", the number of needed iterations has to be indicated). The algorithms check the coherence of the new value with these particularities;
- The new values for the parameters have to be validated: e.g., coherence of the new value and feasibility at runtime (for instance, the "MaxIteration" cannot be adjusted from 20 iterations to 10 iterations if the running polling has already performed 13 iterations), etc.

C. Informational Models for Configurability

The *polling configurability* and its modifiable and non-modifiable parameters have already been introduced in more details in [11] and the main ones are recalled in Figure 4.

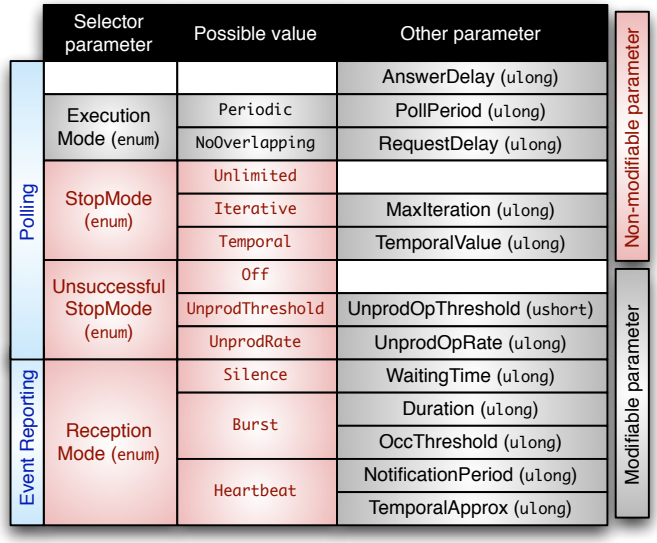


Fig. 4. Configuration Parameters for Monitoring Mechanisms

- The execution mode (`ExecutionMode`) selects the way the polling behaves: the polling operations can be launched either periodically, or in a non-overlapping mode: in the former case, the polling period (`PollPeriod`) parameter defines the periodicity; in the latter one, a time interval (`RequestDelay`) is needed to fix the time between the end of one operation and the beginning of the next one;
- The termination mode (`StopMode`) is the way the polling ends: the polling can indeed be unlimited or limited by a maximum number of occurrences (`MaxIteration`) or by a predefined duration (`TemporalValue`);

 ISBN: 978-1-61208-219-6

$$(M_T, C, Op) \xrightarrow{\mathcal{A}(m_{t_x}, c_x)} (M'_{T'}, C', Op') \text{ with } (M'_{T'}, C', Op') = (M_T, C, Op) \cup \{m_{t_x}, c_x, running\}$$

$$(M_T, C, Op) \xrightarrow{\mathcal{D}(m_{t_x}, c_x)} (M'_{T'}, C', Op') \text{ with } (M'_{T'}, C', Op') = (M_T, C, Op) - \{m_{t_x}, c_x, running\}$$

$$(M_T, C, Op) \xrightarrow{\mathcal{U}(m_{t_x}, c_x)} (M_T, C', Op) \text{ with } C \neq C'$$

$$(M_T, C, Op) \xrightarrow{\mathcal{S}(m_{t_x})} (M_T, C, Op') \text{ with } Op_x = pending$$

$$(M_T, C, Op) \xrightarrow{\mathcal{R}(m_{t_x})} (M_T, C, Op') \text{ with } Op_x = running$$

Fig. 3. Adaptability Operators: Formal Representation

- Let us consider that the polling is iteratively bounded: if a polling operation succeeds before the maximum waiting time (`AnswerDelay`) elapsed, the request is said productive; otherwise, when the answer is lost or delayed, then the request is said unproductive. An autonomous termination mode (`UnsuccessfulStopMode`) can be defined on an unproductive operation maximum rate (`UnprodOpRate`) or on a successive unproductive operation threshold (`UnprodOpThreshold`).

The *event reporting configurability* is more specific. According to some particular issues, it can be interesting to detect how the notifications or events are received (their *behavior*) independently of their *content*. Indeed, the notifications reception frequency can be relevant of any particular issue occurring on the managed system.

Therefore, the reception mode (`ReceptionMode`) parameter has been defined in order to select the management mode of the reception frequency of the notification operations. Three interesting situation profiles have been identified:

- No notification are received during a fixed time interval. This situation is called "silence", and can have two possible interpretations: the system works correctly and no warning or alert have to be raised; or, one (or more) notification(s) was(were) supposed to be received but did not make it, so an issue may have occurred on the system (e.g., the notifier may be on failure). An additional parameter is required to fix the maximum waiting time (`WaitingTime`) of a notification reception;
- Too many notifications are received in a fixed time interval. This behavior is called "burst" and can also be significant of a system issue. Two extra parameters have to be defined to manage it: the duration (`Duration`) on which the calculation is performed and the occurrence threshold (`OccThreshold`) which will trigger the detection of a sporadic behavior;
- The reception of notification operations can be periodic. This situation is called `heartbeat` and is defined with two additional parameters: the notification period (`NotificationPeriod`) and a temporal approximation (`TemporalApprox`) which allows a permitted temporal variation (e.g., due to traffic congestion).

Figure 4 makes a list of the reception mode parameters which describe how the notifications are received by the consumer. Additionally, other typical parameters should be defined to describe how notifications should be produced, filtered and delivered to the consumer when it subscribes to indications or pushed data.

V. Prototype Overview

In order to prove the feasibility of our approach and to measure the overhead due to the adaptation management (e.g., an increased execution time), we have implemented a prototype in an CIM/WBEM environment.

A. Technical Environment

CIM/WBEM are Distributed Management Task Force [4] standards which include a device and service management architecture, an object-oriented information model for describing any type of managed resource and a set of interfaces for accessing them [6]. In our adaptive monitoring service prototype, we provided a CIM representation of our configurability information models and we implemented the adaptability operators using WBEM interfaces. The CIM server is provided by OpenPegasus, an open source implementation of DMTF standards. As for the CIM clients, we used the Java SBLIM API.

1) CIM Server: The CIM/WBEM server is a data supplier and stores into a repository the monitored CIM targets instrumented via the integration modules (gateways between the standard or proprietary monitoring protocols and CIM). It also enables the storage and the manipulation of the configuration parameters included in the monitoring class. Finally, it enables the notifications of CIM indications toward the subscribers.

Therefore, the configurability level has been instrumented on a CIM server, which is executed on a virtual machine running Linux Ubuntu TLS 8.04 "Hardy Heron". The virtualization system is VirtualBox r73507.

2) CIM client: The CIM/WBEM client takes the configuration from the server initially and when a readjustment is achieved. It acts too as a WBEM listener and subscribes on the server in order to receive every CIM indications relative to an update of a CIM instance of the monitoring class.

The modification of the monitoring mechanisms (adaptability level) implies the use of a CIM client. It allows the dynamic management of the proposed CIM models by the adjunction of operators to see, modify, add and delete CIM objects and associations. The adaptability level is then running on the physical machine hosting the virtual machine. This physical machine runs Mac OS X 10.5.8 "Leopard".

 ISBN: 978-1-61208-219-6

B. Temporal Overhead due to Adaptation Management

All the presented first results have been obtained by computing the average of a hundred measures. The host machine runs the CIM client and the virtual machine runs the CIM server. The needed CIM objects (targets, polling) are the only objects registered onto the CIM repository. The polling mechanism is the only mechanism that has been measured so far. The measures lead to a preliminary analysis, which will have to be enhanced.

Figure 5 shows the execution duration (in milliseconds) needed for the execution of each adaptability operator in the case of a polling mechanism (the only mechanism measured so far). These measures are the following:

ADD	699 ms
DELETE	1004 ms
RESUME	353 ms
SUSPEND	110 ms
UPDATE	
– selector parameter	141 ms
– sub-parameter	193 ms
– global update	150 ms
– target adjunction	374 ms
– target deletion	304 ms

Fig. 5. Execution Duration for Adaptability Operators Execution

- ADD: the duration between the time when the $\mathcal{A}$ operator is invoked (in order to add a two targets polling operation) and the time when the polling operationally begins;
- SUSPEND: the duration between the time when the $\mathcal{S}$ operator is invoked (suspension of the polling operation) and the time when the polling is operationally suspended;
- RESUME: the duration between the time when the $\mathcal{R}$ operator is invoked (resumption of the previously suspended polling operation) and the time when the polling is operationally resumed (according to the context registered while it was suspended);
- UPDATE: the duration between the time when the $\mathcal{U}$ operator is invoked and the time when the polling takes into account the modification value. Five kinds of update are measured:
  - Selector parameter: the modification of the value of the `ExecutionMode` parameter;
  - Sub-parameter: the modification of any other modifiable configuration parameters. It implies a test on the relative selector parameter: e.g., the polling period can be modified only if the execution mode selector parameter is set to “Periodic”;
  - Global update: a global modification of the polling configuration: more than one parameter will be updated;
  - Target adjunction: the adjunction of a third target to the polling;
  - Target deletion: the deletion of the third target of the polling;
- DELETE: the duration between the time when the $\mathcal{D}$ operator is invoked (deletion of the two targets polling operation of the CIM server) and the time when the polling is totally removed from the CIM server.

As shown in Figure 5, the execution duration of a polling deletion is the longest execution duration (1004 ms), followed by the polling adjunction (699 ms). From this, it appears that the suspension and resumption operators are really interesting when a polling has been added: a combined suspension and resumption lasts 463 ms, while a combined deletion and adjunction lasts 1703 ms.

An adaptation at runtime can be enforced fastly:

- A selector parameter (the execution time) can be modified in 141 ms;
- A sub-parameter can be modified in 193 ms: the extra time is indeed due to a test over the selector parameter to check if the parameter can be modified for consistency ensuring purpose;
- A global update (the modification of the whole polling configuration) lasts 150 ms;
- The adjunction and deletion of a polling target lasts 374 ms and 304 ms, respectively.

VI. Conclusion and Future Work

This paper presented a generic monitoring framework that aims at enforcing the monitoring adaptation at runtime and without any disruption. When the adaptation is performed, the monitoring is made the less intrusive possible, by efficiently adjusting itself to every situation variation (business objectives changes, monitoring self-optimization, constraints variations on resources).

Following a control-oriented viewpoint, three capabilities have been identified within the framework: configurability, adaptability and governability. To ensure configurability, we have defined, for each type of basic mechanisms, generic configuration parameters: some are modifiable, the others are not. These specifications are independent of any management protocol; the configuration has been integrated on a CIM server. To ensure adaptability, the adaptation operators have been formalized and implemented on a CIM client thanks to the Java SBLIM API.

The resulting framework is a basis for the decision-making level as it is generic enough to support any solutions providing adaptation decisions.

Future works will tend now to integrate the governability capability to the presented framework. This framework is a basis which support monitoring adaptation enforcement. Integrating governability means controling the adaptation decision. A choice has to be done to determinate what is the best solution to enforce the governability capability, according to the underlying framework and needs. Moreover, we intend to test the complete prototype in a real managed environment.

References

[1] L. Chung and N. Subramanian, ”Adaptable Architecture Generation for Embedded Systems,” Journal of Systems and Software, 71(3), (2004), pp. 271–295.

 ISBN: 978-1-61208-219-6

[2] DMTF, "Common Information Model (CIM) Infrastructure" specification v.2.6.0 [online], (2010) [retrieved: May, 2012], http://www.dmtf.org/standards/cim.
[3] M. Dilman and D. Raz., "Efficient Reactive Monitoring," Infocom 2001, 2, (April 2001), pp. 1012–1019.
[4] DMTF Standards body, "Distributed Management Task Force, Inc." [online], (2012) [retrieved: May, 2012], http://www.dmtf.org.
[5] E.P. Duarte Jr., M.A. Musicante, and H.H. Fernandes, "ANEMONA: a Programming Language for Network Monitoring Applications," Internat. Journ. of Net. Managem., 18(4), (August 2008).
[6] Chris Hobbs: "A Practical Approach to WBEM/CIM Management," Auerbach, 2004, 344p.
[7] J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," Computer, 36(1), (January 2003), pp. 41–50.
[8] A. Lahmadi, "Performances des fonctions et architectures de supervision de réseaux et de services," Doctorat de l'Université Nancy II, (Dec. 2007).
[9] M.L. Massie, B.N. Chun, and D.E. Culler, "The Ganglia Distributed Monitoring System: Design, Implementation and Experience," Parallel Computing, 30(7), (2004), pp. 817–840.
[10] P. Moghé, and M.H. Evangelista, "RAP – Rate Adaptive Polling for Network Management Applications," in Proceedings of IEEE NOMS'98, (1998), pp. 395–399.
[11] A. Moui, T. Desprats, E. Lavinal, and M. Sibilla, "Managing Polling Adaptability in a CIM/WBEM Infrastructure," Internat. DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and New Technologies (SVM10), (2010), pp. 1–6.
[12] A. Moui, and T. Desprats, "Towards Self-Adaptive Monitoring Framework for Integrated Management," 5th IFIP Internat. Conference on Autonomous Infrastructure, Management and Security (AIMS 2011), (2011), pp. 160–163.
[13] A.G. Prieto, and R. Stadler, "Controlling Performance Trade-offs in Adaptive Network Monitoring," Proceedings of IM 2009, (2009), pp. 359–366.
[14] N. Samaan and N. Karmouch, "Towards Autonomic Network Management: an Analysis of Current and Future Research Directions," Communications Surveys & Tutorials, IEEE, 11(3), (2009), pp. 22–36.
[15] DMTF: "Web-Based Enterprise Management" specifications [online], (2012) [retrieved: May, 2012], http://www.dmtf.org/standards/wbem.
[16] Z. Xue, X. Dong, and W. Wu, "AOCMS: An Adaptive and Scalable Monitoring System for Large-Scale Clusters," Proc. APSCC, (2006), pp. 466–472.

 ISBN: 978-1-61208-219-6

Improvement of the Calibration Process for Class E_1 Weights Using an Adaptive Subdivision Method

Adriana Vâlcu/ National Institute of Metrology
Mass Laboratory
INM
Bucharest, Romania
e-mail: adriana.valcu@inm.ro; adivaro@yahoo.com

***Abstract*— Taking into account the amount and variety of measurements involved in scientific, industrial and legal activities that need traceability to the national mass standards of each country, it can be considered that mass standards calibration is one of the most important activities of the National Metrology Institutes (NMIs). For the determination of the conventional mass, in the calibration of weights of the highest accuracy classes, the subdivision method and its variants are widely used. For the NMIs, it is very important to demonstrate and maintain their capability of applying with good results such methods. In this respect, a calibration procedure for the determination of conventional mass, called "adaptive subdivision method" was developed in the Mass Laboratory of the Romanian National Institute of Metrology, which can lead to an improvement of CMCs (Calibration and Measurement Capabilities, approved and published in the BIPM database). According to the International Recommendation OIML R 111, the weights of nominal values greater than 1 g may have a cylindrical shape with a lifting knob. Considering this kind of shape and the use of an automatic comparator, with the maximum capacity of 1 kg, the diameter of the weighing pan is too small for placing a group of weights in the range of (500…100) g; therefore, the usual subdivision method can not be applied for the calibration of weights. The "adaptive subdivision method", presented in this paper, allows the cylindrical weights with a lifting knob, having nominal values of (500…100) g, to be calibrated using an automatic comparator (which is not equipped with weight support plates). The method can be used for class E_1 weights, where the highest accuracy is required. In this case, the resulting calibration uncertainty for the unknown weights is better than that usually obtained for E_1 masses, being at the level of reference standards.**

Keywords - Subdivision method; automatic comparator; efficiency of design.

I. INTRODUCTION

In 1889, at the First Conference of Weights and Measures (CGPM), the kilograms prototypes were shared - by chance - for each country. Romania has received the "National kilogram Prototype No. 2" (NPK).

NPK is a solid cylinder of Platinum-Iridium alloy (90%, 10%), having a height equal to its diameter (39 mm). Now, it is maintained by the National Institute of Metrology and serves as a reference for the entire dissemination of the mass unit in Romania

The realization and dissemination of the unit of mass by the Mass Laboratory of the Romanian National Institute of Metrology starts from the reference stainless steel standards (a set of three 1 kg mass standards and two sets of disc weights from 500 g to 50 g), which are traceable to the International Prototype Kilogram through the mass of the Romanian Prototype Kilogram No 2.

Starting from these reference stainless steel standards, submultiples and multiples of the unit are realized to permit the masses of additional bodies to be determined with traceability to the international standard. This takes place with the aid of several weights sets E_1 of suitable grading (in most cases 1, 2, 2, 5) which are determined "in themselves" according to proper weighing designs and by using a least squares analysis (with subdivision or multiplication methods).

In the calibration of class E_1 weights, when the highest accuracy is required, the subdivision method is mainly used.

The subdivision weighing design has both advantages and disadvantages:

- advantages [2]:

a) minimizes handling (and hence wear) of standards;

b) produces a set of data providing important statistical information about the measurements and the daily performance of the individual balances;

c) offers a redundancy of data.

- disadvantages [2]:

a) requires a relatively complex algorithm to analyze the data (as compared with other methods, for example Borda [3]);

b) necessitates placing groups of weights on the balance pans (this can cause problems for instruments with poor eccentricity characteristics, or automatic comparators designed to compare single weights).

To apply the calibration by subdivision method on the automatic comparator, a set of disc weights (reference standards) has been used. These weights constitute both support plates and check standards.

The main objective in the search for better designs was to find a calibration scheme which can be performed considering the following factors: the automatic comparator, the diameter of the disc weights (so that a group of OIML weights can be placed over) and the efficiency of design matrix.

 ISBN: 978-1-61208-219-6

The article is divided into 6 sections as follows: introduction, equipments and standards used in calibrations, statistical tools for evaluation of the measurement process and mass determination, analysis of uncertainties, quality assessment of the calibration, conclusions.

II. EQUIPMENTS AND STANDARDS USED IN CALIBRATION

The weighing system includes a proper balance (mass comparator) with weights transporter, a monitoring system of environmental conditions and a MC Link software

The mass comparator used was an automatic one, with the following specifications:

- maximum capacity: 1011 g;
- readability: 0.001 mg;
- pooled standard deviation: (0.4 to 2) µg (for nominal masses 100 g to 1 kg, respectively).

For accurate determination of the air density an environmental conditions monitoring system was used, consisting in a precise "climate station", model Klimet A30.

Technical parameters for Klimet A30 are:

- temperature: readability : 0.001°C;
  U (k=2) : 0.03°C;
- dew point: resolution : 0.01°C;
  U (k=2) : 0.05°C;
- barometric pressure: resolution : 0.01 hPa;
  U (k=2) : 0.03 hPa;

The mass standard used for the comparisons was an 1 kg reference standard, Ni 81, Fig 1, whose mass value was determined at BIPM.

Fig.1. Reference standard of 1 kg, Ni81

Ni 81 had been purchased by the National Institute of Metrology in 1981. This mass standard is the second in importance after the NPK. The data included in its calibration certificate are as follows:

m_{Ni81} = 1 kg + 0.130 mg, U = 0.028 mg , (k=2);

The weights involved in calibration are:

- unknown E_1 weights (from 500g to 100g, marked with A12…A9) having OIML shape, Fig 2.

Fig. 2. Weights of class E_1

- disc weights (reference weights, marked with NA), Fig 3.

Fig. 3. Reference disc weights

For all the weights, the volumes *V* and associated uncertainties *U(V)*are given in their calibration certificates.

Table I shows these values:

TABLE I. VOLUMES *V* AND ASSOCIATED UNCERTAINTIES *U(V)* OF THE WEIGHTS

Nominal mass g	**Marking**	***V*** cm^3	***U*(*V*)** **cm^3**
1000 ref	Ni	127.7398	0.0012
500	NA	62.5480	0.0007
500	A12	62.266	0.032
200	A11	24.853	0.008
200	A10	24.853	0.008
100	NA	12.5083	0.0005
100	A9	12.456	0.004

III. STATISTICAL TOOLS FOR EVALUATION OF THE MEASUREMENT PROCESS AND MASS DETERMINATION

A. Method used to evaluate the efficiency of the weighing design

The dissemination of the mass scale to E_1 weights, using a single reference standard, requires mass comparisons between weights and groups of weights.

A mass calibration design (or design matrix) describes the general setup of these comparisons.

For a given number of mass comparisons, a criterion for the choice of a design matrix is that, the variances of the estimates be as small as practicable [4].

For this reason, the idea of efficiency was introduced, to enable designs to be analyzed using this criterion, taking into account the variances of the weighing results.

The efficiency is very useful when comparing designs involving the same masses and balances, even if the number of mass comparisons differs. It is desirable that the efficiency of a design be large, as this would indicate that the variances are small [4].

Table II lists the mass comparisons possible for the 1 kg to 100g decade, taking into account the following elements: the automatic comparator and the diameter of the disc weights (so that a group of OIML weights can be placed over).

 ISBN: 978-1-61208-219-6

TABLE II. POSSIBLE MASS COMPARISONS FOR THE 1 kg TO 100g DECADE

Obs. No	Mass						
	Ni 81	500NA	500A12	200A11	200A10	100NA	100A9
1	-1	1	1	0	0	0	0
2	-1	1	0	1	1	1	0
3	-1	1	0	1	1	0	1
4	0	1	-1	0	0	0	0
5	0	1	0	-1	-1	-1	0
6	0	0	1	-1	-1	-1	0
7	0	0	0	1	-1	-1	1
8	0	0	0	-1	1	-1	1
9	0	0	0	1	-1	0	0
10	0	0	0	1	0	-1	-1
11	0	0	0	0	1	-1	-1
12	0	0	0	0	0	1	-1

To establish the design matrix „X" of the comparisons, several versions were performed, then calculating the efficiency of the design for each of them.

For example, using the notation of [4], for the design (2, 1, 1, 2, 0, 1, 1, 0, 1, 1, 2, 1) an efficiency of 0.38 was obtained, while for the design (1, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1) the efficiency obtained was 0.61.

Finally, the design (2, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1) was chosen, having 13 equations of condition, since the value for the efficiency was greater, namely 1.04.

The efficiency was calculated in the following manner. Once all weighing are completed, the first step is to form the design matrix, "X", which contains the information on the equations used (the weighing design).

Entries of the design matrix are +1, –1, and 0, according to the role played by each of the parameters (from the vector β) in each comparison. Symbols used:

X the format for matrix: $X = (x_{ij})$; i=1… n;
j = 1,…, k; x_{ij} = 1, –1 or 0;
β vector of unknown departures (β_j);
s vector containing the standard deviation of each comparison;
Y the vector of measured values "y_i", including buoyancy corrections according to (6) .

$$X=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad s=\begin{bmatrix} \mathbf{0.016} \\ 0.0013 \\ 0.0013 \\ 0.0009 \\ 0.0010 \\ 0.0017 \\ 0.0017 \\ 0.0004 \\ 0.0013 \\ 0.0006 \\ 0.0005 \\ 0.0005 \\ 0.0007 \\ 0.0009 \end{bmatrix} mg \quad Y=\begin{bmatrix} \mathbf{-3.1583} \\ 3.1896 \\ 3.1896 \\ 3.0994 \\ 3.0758 \\ 0.1001 \\ 0.1001 \\ 0.1796 \\ 0.0801 \\ -0.0052 \\ -0.0396 \\ -0.0414 \\ -0.0579 \\ 0.0225 \end{bmatrix} mg \quad \beta=\begin{bmatrix} \text{Ni81} \\ \text{500NA} \\ \text{500A12} \\ \text{200A11} \\ \text{200A10} \\ \text{100NA} \\ \text{100A9} \end{bmatrix} \quad (1)$$

where:
"Ni81" represents the reference kilogram standard;
"NA" the disc weights;
"A12, A11, A10, A9" OIML weights of E_1 class.

In the "Fig. 4" it can be seen a detail of the weights combination: 500NA+200A12+200A11+100A9, part of determination "y_4"

Fig. 4. The combination of the weights from the 4th determination

The observations vector Y has a diagonal variance - covariance matrix G:

$$\boldsymbol{G} = \text{diag}\,(u_r^2, s_1^2, s_2^2, \ldots s_{n-1}^2) \quad (2)$$

where u_r^2, is the square of the uncertainty of reference standard, named Ni81, and s_j^2 (j= 1, . . . ***n-*** 1) is the variance of the j-th comparison.

If G' is the same as $\boldsymbol{G}$ without the first row and column, the matrix $G^{'-1/2}$ can be calculated.

By denoting with $\boldsymbol{J}$ a (n-1) x (k-1) a sub-design matrix that would be used if the same mass comparisons are carried out, without the use of a reference mass, the matrix K can be defined:

$$K = G^{'-1/2}\, J \quad (3)$$

Calculating K^T, which is transpose of K, one can determine the inverse $(K^T \bullet K)^{-1}$:

$$\left(K^T \cdot K\right)^{-1} = \begin{bmatrix} \mathbf{0.120} & -0.011 & 0.016 & 0.019 & 0.019 & -0.001 \\ -0.011 & \mathbf{0.413} & 0.009 & 0.009 & 0.005 & 0.004 \\ 0.016 & 0.009 & \mathbf{0.059} & 0.001 & 0.009 & 0.014 \\ 0.019 & 0.009 & 0.001 & \mathbf{0.063} & 0.009 & -0.004 \\ 0.019 & 0.005 & 0.009 & 0.009 & \mathbf{0.054} & 0.003 \\ -0.001 & 0.004 & 0.014 & -0.004 & 0.003 & \mathbf{0.071} \end{bmatrix} \quad (4)$$

If v_i are the diagonal elements of $(K^T \cdot K)^{-1}$ corresponding to the i^{-th} mass, σ_m is the largest of the σ_i, then the efficiency of the design, represented by the matrix $\boldsymbol{X}$ is defined as [4]:

$$E = \sum v_i^{-1} \cdot h_i^2 \cdot \sigma_m^2 \,/\, (n-1) \quad (5)$$

n is the number of comparisons;
h_i the ratio between the nominal values of the unknown weights and the reference.

In Table III and Table IV, the calculation of the efficiency for different designs containing 13 equations of condition is presented.

 ISBN: 978-1-61208-219-6

TABLE III. THE CALCULATION OF EFFICENCY FOR THE DESIGN

(2, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1)

$1/v_i$	h	$h_i^2 \cdot 1/v_i$	n-1	σ^2_m	$(h_i^2 \cdot 1/v_i) \cdot \sigma^2_m/(n-1)$	Standard deviation (μg)
8.33	0.5	2.0819	12	2.89	0.501	0.35
2.42	0.5	0.606			0.146	0.64
16.80	0.2	0.672			0.162	0.24
15.82	0.2	0.633			0.152	0.25
18.55	0.1	0.185			0.045	0.23
14.07	0.1	0.141			0.034	0.27
					E = 1.04	

TABLE IV. THE CALCULATION OF EFFICENCY FOR THE DESIGN

(2, 1, 1, 2, 0, 1, 1, 0,1, 1, 2, 1)

$1/v_i$	h	$h_i^2 \cdot 1/v_i$	n-1	σ^2_m	$(h_i^2 \cdot 1/v_i) \cdot \sigma^2_m/(n-1)$	Standard deviation (μg)
1.946	0.5	0.487	12	2,89	0,117	0,72
1.946	0.5	0.487			0,117	0,72
5.773	0.2	0.231			0,056	0,42
5.222	0.2	0.209			0,050	0,44
8.848	0.1	0.088			0,021	0,34
7.410	0.1	0.074			0,018	0,37
					E = 0.38	

It can be seen that, in the first case (Table III), a higher efficiency was obtained, which indicates that the standard deviations are smaller. Therefore, this weighing design was finally chosen to calculate the mass of the unknown and uncertainty of calibration.

B. Mass results obtained in the calibration of weights

If it is denoted by (A) the weighing of the reference weight and (B) the weighing of the test weight, an ABBA weighing cycle represent the sequence in which the two weights are measured to determine the mass difference of a comparison in a design matrix.

The calibration data used are obtained from the weighing cycles ABBA for each y_i (which is the weighing comparison according to design matrix "*X*").

The general mathematical model for "*y*", corrected for air buoyancy is:

$$y = \Delta m + (\rho_a - \rho_o)(V_1 - V_2) \quad (6)$$

with:

Δm is the difference of balance readings;

ρ_o 1.2 kg $\cdot m^{-3}$ the reference air density;

ρ_a air density at the time of the weighing;

V_1, V_2 volumes of the weights (or the total volume of each group of weights) involved in measurement.

To estimate the unknown masses of the weights, the least square method was used [4, 5, 6].

The design matrix "*X*" and the vector of observations "*Y* " are transformed (to render them of equal variance) in *U* and *W* respectively, as follows [4]:

$$\boldsymbol{U} = \boldsymbol{G}^{-1/2}\boldsymbol{X} \quad \text{and} \quad \boldsymbol{W} = \boldsymbol{G}^{-1/2}\,Y \quad (7)$$

$$U = \begin{bmatrix} 0.0625 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.769 & 0.769 & 0,769 & 0 & 0 & 0 & 0 \\ -0.769 & 0.769 & 0,769 & 0 & 0 & 0 & 0 \\ -1.111 & 1.111 & 0 & 1,111 & 1,111 & 1,111 & 0 \\ -1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0.588 & -0.588 & 0 & 0 & 0 & 0 \\ 0 & 0.588 & -0.588 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.769 & -0.769 & -0.769 & -0.769 & 0 \\ 0 & 2.50 & 0 & -2.5 & -2.5 & -2.5 & 0 \\ 0 & 0 & 0 & 1.667 & -1.667 & -1.667 & 1.667 \\ 0 & 0 & 0 & -2 & 2 & -2 & 2 \\ 0 & 0 & 0 & 2 & 0 & -2 & -2 \\ 0 & 0 & 0 & 0 & 1.429 & -1.429 & -1.429 \\ 0 & 0 & 0 & 0 & 0 & 1.111 & -1.111 \end{bmatrix} \quad W = \begin{bmatrix} -0.1974 \\ 2.4535 \\ 2.4535 \\ 3.4438 \\ 3.0758 \\ 0.0589 \\ 0.0589 \\ 0.0616 \\ 0.4490 \\ -0.0087 \\ -0.0792 \\ -0.0828 \\ -0.0827 \\ 0.0250 \end{bmatrix} mg \quad (8)$$

The estimates β_j and their variance-covariance matrix $V_{\beta j}$ are calculated as follows:

$$\langle \beta_j \rangle = (U^T \cdot U)^{-1} \cdot U^T \cdot W = \begin{bmatrix} \mathbf{Ni81} \\ 500NA \\ 500A12 \\ 200A11 \\ 200A10 \\ 100NA \\ 100A9 \end{bmatrix} = \begin{bmatrix} \mathbf{-3.158} \\ 0.0615 \\ -0.0345 \\ -0.0534 \\ -0.0704 \\ 0.0053 \\ -0.0175 \end{bmatrix} mg \quad (9)$$

$$V_{\beta j} = \left(U^t \cdot U\right)^{-1} = \begin{bmatrix} \mathbf{256} & 128 & 128 & 51 & 51 & 26 & 26 \\ & \mathbf{64} & 64 & 26 & 26 & 13 & 13 \\ & & \mathbf{64} & 26 & 26 & 13 & 13 \\ & & & \mathbf{10} & 10 & 5 & 5 \\ & & & & \mathbf{10} & 5 & 5 \\ & & & & & \mathbf{3} & 3 \\ & & & & & & \mathbf{3} \end{bmatrix} \mu g^2 \quad (10)$$

The diagonals elements V_{jj} , of the V_{β}j represent the variance of the weights (which includes the type A variance combined with the variance associated to reference standard).

IV. ANALYSIS OF UNCERTAINTIES

A. Uncertainty of the weighing process, u_A

The variance $V_{\beta j}$ can be also expressed as [4]:

$$V_{\beta j} = h\, h^T \cdot \sigma_r^2 + R \quad \text{with} \quad R = \begin{pmatrix} 0 & \vdots & 0^T \\ \cdots & \cdots & \cdots \\ & \vdots & \left(K^T \cdot K\right)^{-1} \end{pmatrix} \quad (11)$$

The diagonals elements of the $(K^T \bullet K)^{-1}$ represents the type *A* variance of the unknown weight. From here, the type *A* standard uncertainty can be obtained:

$$u_{A(\beta_j)} = \begin{bmatrix} 0.35 \\ 0.64 \\ 0.24 \\ 0.25 \\ 0.23 \\ 0.27 \end{bmatrix} \mu g \quad (12)$$

 ISBN: 978-1-61208-219-6

B. Type B uncertainty

The components of type B uncertainties [1,9] are:

1) Uncertainty associated with the reference standard, u_r, for each weight is given by [1]:

$$u_{\mathbf{r}(\beta_j)} = h_{\mathbf{j}} \cdot u_{\mathbf{r}} = h_{\mathbf{j}} \cdot \sqrt{u_{\mathbf{cert}}^2 + u_{\mathbf{stab}}^2} = \begin{bmatrix} 0.008 \\ 0.008 \\ 0.0032 \\ 0.0032 \\ 0.0016 \\ 0.0016 \end{bmatrix} mg \quad (13)$$

where:

u_{cert} uncertainty of the reference standard from the calibration certificate;

u_{stab} uncertainty associated with stability of reference standard.

2) Uncertainty associated with the air buoyancy corrections, u_b is given by [1]*:*

$$u_{b(\beta j)}^2 = (V_j - V_r h_j)^2 u_{\rho a}^2 + (\rho_a - \rho_o)^2 u_{Vj}^2 + [(\rho_a - \rho_o)^2 - 2(\rho_a - \rho_o)(\rho_{a1} - \rho_o)] u_{Vr}^2 h_j \quad (14)$$

where:

V_j, V_r represents the volume of test weight and reference standard, respectively;

ρ_a air density at the time of the weighing;

$u_{\rho a}$ uncertainty for the air density determined at the time of the weighing, calculated according to CIPM formula;

ρ_o = 1,2 kg·m^{-3} is the reference air density;

u_{Vj}^2, u_{Vr}^2 uncertainty of the volume of test weight and one of the reference standard, respectively;

ρ_{a1} air density determined from the previous calibration of the standard.

Uncertainty associated with the air buoyancy corrections, u_b, calculated for each weight is:

$$u_{b(\beta_j)} = \begin{bmatrix} 2.6 \cdot 10^{-4} \\ 4.5 \cdot 10^{-4} \\ 1.7 \cdot 10^{-4} \\ 1.7 \cdot 10^{-4} \\ 5.3 \cdot 10^{-5} \\ 7.7 \cdot 10^{-5} \end{bmatrix} mg \quad (15)$$

3) Uncertainty due to the sensitivity of the balance

When the balance is calibrated with a sensitivity weight (or weights) of mass, m_s, and standard uncertainty, $u_{(ms)}$, the uncertainty contribution due to sensitivity is [1]:

$$u_s^2 = \Delta m_c^2 \cdot [u_{ms}^2 / m_s^2 + u_{(\Delta Is)}^2 / \Delta I_s^2] \quad (16)$$

where:

ΔI_s the change in the indication of the balance due to the sensitivity weight;

$u(\Delta I_s)$ the uncertainty of ΔIs;

Δm_c the average mass difference between the test weight and the reference weight.

Usually, the term from brackets is taken from the calibration certificate of the mass comparator.

Uncertainty associated to the sensitivity of the balance is calculated, giving:

$$u_s = \begin{bmatrix} 7 \cdot 10^{-7} \\ 7 \cdot 10^{-7} \\ 2 \cdot 10^{-7} \\ 2 \cdot 10^{-7} \\ 9 \cdot 10^{-8} \\ 9 \cdot 10^{-8} \end{bmatrix} mg \quad (17)$$

4) Uncertainty associated with the display resolution of the balance, u_{rez}, (for electronic balances) is calculated according to the formula [1]:

$$u_{rez} = \left(\frac{d/2}{\sqrt{3}} \right) \times \sqrt{2} = 0.00041 mg \quad (18)$$

C. Combined standard uncertainty

The combined standard uncertainty of the conventional mass of the weight βj is given by [1]:

$$u_{c(\beta j)} = [(u_A^2(\beta j) + u_r^2(\beta j) + u_b^2(\beta j) + u_s^2 + u_{rez}^2]^{1/2} \quad (19)$$

D. Expanded uncertainty

The expanded uncertainty "*U*" of the conventional mass of the weights β_j is given by:

$$U_{(\beta j)} = 2 \cdot u_c(\beta j) = \begin{bmatrix} 500NA \\ 500E_1 \\ 200NA \\ 200E_1 \\ 100NA \\ 100E_1 \end{bmatrix} = 2 \cdot \begin{bmatrix} 0.0080 \\ 0.0080 \\ 0.0032 \\ 0.0032 \\ 0.0017 \\ 0.0017 \end{bmatrix} = \begin{bmatrix} 0.016 \\ 0.016 \\ 0.006 \\ 0.006 \\ 0.003 \\ 0.003 \end{bmatrix} mg \quad (20)$$

V. QUALITY ASSESSMENT OF THE CALIBRATION

As shown, for calibration of the E_1 weights disc weights of 500 g and 100 g were used, having both the role of check standards and weight support plates for the whole determination.

To see if the mass values obtained for disc weights are consistent with previous values, it is necessary to perform a statistical control. The purpose of the check standard is to assure the validity of individual calibrations. A history of values on the check standard is required for this purpose [1]. Considering that for the disc weights there are no sufficient calibration data to perform a statistical control according to [1], the method of normalized error E_n was chosen, which takes into account the result and its uncertainty from the last calibration.

 ISBN: 978-1-61208-219-6

The results obtained for the disc weights in this subdivision procedure are compared with data from their calibration certificates [7, 9]. The differences in values are normalized using the formula [8]:

$$E_n = \frac{\delta_{subdiv} - \delta_{certif}}{\sqrt{U^2_{subdiv} + U^2_{certif}}} \quad (21)$$

where:

δ_{subdiv} represents the mass error of the disc weight obtained by subdivision method;

δ_{certif} the mass error of the disc weight from the calibration certificate;

U_{subdiv} the expanded uncertainty of the disc weight obtained in subdivision method;

U_{certif} the expanded uncertainty from the calibration certificate of the disc weight.

Using this formula, the measurement and the reported uncertainty are acceptable if the value of E_n, is between -1 and +1.

Table V presents the results obtained for the normalized errors, E_n.

TABLE V. COMPARISON OF MEASUREMENT RESULTS OF DISC WEIGHTS, OBTAINED BY SUBDIVISION METHOD AND RESULTS FROM THE CALIBRATION CERTIFICATE

Nominal mass of disc weight	Subdivision		Calibration certificate		E_n
g	*δ (mg)*	*U (mg)*	*δ (mg)*	*U (mg)*	
500NA	0.062	0.016	0.076	0.017	0. 6
100NA	0.005	0.003	0.008	0.004	0..5

VI. CONCLUSIONS

An evaluation procedure has been presented, used for the calibration of a set of weights by subdivision (similar considerations had been published by the author in [9]). This calibration procedure for the determination of conventional mass of the weights was developed in the Mass Laboratory of the National Institute of Metrology, and can lead to an improvement of CMCs (Calibration and Measurement Capabilities), approved and published in the BIPM database.

The main feature of this kilogram subdivision method is represented by the fact that the calibration of the weights (whose shape is in accordance with OIML R111) is performed using an automatic mass comparator. Uncertainties obtained using this method for the unknown weights are better than those usually occur for E_1 (when only manual measurements are possible): 0.060 mg for the 500 g weight, 0.03 mg for the 200 g and 0.017 mg for the 100 g, being at the level obtained for reference standards (marked with NA).

The comparison of results obtained for the disc weights by the subdivision method with those from the calibration certificate using the normalized errors E_n, confirms the consistency of the results.

The method described in this paper for calibration of E_1 weights can be used when the highest accuracy is required.

REFERENCES

[1] OIML, International Recommendation No 111,"Weights of classes E1, E2, F1, F2, M1, M2, M3", 2004, pp. 65-69.

[2] S Davidson, M Perkin, M Buckley, "Measurement Good Practice Guide No 71", NPL, TW11 0LW, June 2004, pp. 8-9.

[3] L. Wiener, S. Hilohi, A.Nadolo, Lucia Lazeanu, "Tehnica Măsurării Maselor, Volumelor şi Mărimilor Analitice ("Technique of masse measurement, volumes and analytical quantities"), 1977, pp. 45-46.

[4] E.C.Morris, "Decade design for weighings of Non-uniform Variance", Metrologia 29, 1992, pp. 374-375.

[5] A.Valcu, "Test procedures for class E1 weights at the Romanian National Institute of Metrology. Calibration of mass standards by subdivision of the kilogram", Bulletin OIML, Vol. XLII, No. 3, July 2001, pp. 11-16.

[6] Schwartz R ,"Guide to mass determination with high accuracy", PTB –MA-40, 1995, pp. 54-58.

[7] Matej Grum, Matjaž Oblak, Ivan Bajsić and Mihael Perman,. "Subdivision of the unit of mass using weight support plates", Proceedings of XVII IMEKO World Congress, Croatia, 2003, pp. 407-408.

[8] International Standard ISO 13528:2005-09 (E), "Statistical methods for use in proficiency testing by interlaboratory comparisons", 2005, pp. 27-28.

[9] Adriana Vâlcu and Dumitru Dinu, "Subdivision method applied for OIML weights using an automatic comparator", Proceedings of XIX IMEKO World Congres, Lisbon, 2009, pp. 281-283.

 ISBN: 978-1-61208-219-6

Leveraging the Ubiquitous Web as a Secure Context-aware Platform for Adaptive Applications

Heiko Desruelle, Frank Gielen
Dept. of Information Technology – IBCN
Ghent University – IBBT
Ghent, Belgium
{*heiko.desruelle, frank.gielen*}*@intec.ugent.be*

John Lyle
Dept. of Computer Science
University of Oxford
Oxford, UK
john.lyle@cs.ox.ac.uk

***Abstract*—The availability as well as diversity of connected devices has turned the Internet into a ubiquitous concept. In addition to desktop and laptop PCs, the Internet also connects numerous mobile devices, home entertainment systems, and even in-car units. Various new types software applications arise, trying to make optimal use of this trend. However, as the fragmentation of devices and platforms grows, application developers are increasingly facing the need to cover a wider variety of target devices. Maintaining a viable balance between development costs and market coverage has turned out to be a challenging issue when developing applications for such a ubiquitous ecosystem. In this paper, we present the Webinos approach, a distributed web runtime that adaptively leverages the device-independent characteristics of the Web. By introducing the concept of context-aware Personal Zones, the Webinos platform aims to facilitate the development of self-adaptive and immersive applications, optimized for ubiquitous computing environments.**

***Keywords*-ubiquitous web; context-aware platform; distributed runtime; adaptive applications; webinos**

I. INTRODUCTION

The Internet is drastically changing the way people work and live. As the diversity of connected devices is increasing rapidly, the Internet is penetrating our everyday lives through a multitude of devices. From desktop and laptop PCs, to mobile devices, to home entertainment systems and even in-car headunits, users throughout all consumer segments should prepare for a connected experience [1]. The evolution towards a ubiquitous Internet creates the opportunity for numerous new and innovative software applications. The main driver for such applications would be to seamlessly enable the inherently nomadic character of a ubiquitous system. Furthermore, this driver should aim to enable users to access and share information whenever and wherever they want, regardless of the device type that is being used to initiate the operation.

The development and deployment of applications for such a ubiquitous ecosystem, however, introduces an important series of resource-consuming requirements [2]. The available combinations of hardware characteristics, operating systems, software frameworks, etc. are virtually endless. For software developers, this diversity has turned out to be a double-barreled asset. It provides consumers the freedom to operate applications at will across several devices. On the other hand, the device diversity asset heavily fragments the application's delivery targets. By the absence of a general native development solution, developers often have no alternative than to create and maintain a set of device-dependent versions of their applications. Hence, ensuring a viable balance between development costs and an application's market coverage will more than ever become a challenging issue.

Against this backdrop, the use of web technologies for application development purposes has proven to be a viable long-term candidate solution [3]. Through years of standardization efforts and the wide adoption of languages such as HTML, CSS, and JavaScript, the web can be deployed as a powerful foundation for universal application development and delivery. Running on top of the Internet infrastructure, the web application ideology is rapidly gaining momentum amongst developers.

A web-based application development approach has been explored from various perspectives. Developers can opt for pure web applications, running in a standard browser environment. However, due to the sandboxed nature of browsers this approach drastically limits the available APIs (Application Programming Interfaces) to the underlying device. In turn, a hybrid web application approach was introduced providing developers access to a richer API set, whilst still maintaining most of the cross-platform advantages from pure web applications. This type of application is still built using web technology, but no longer uses the browser as the client-side runtime environment. A separate client-side web runtime framework is deployed to bridge the gap between native and web applications by granting the application scripting access to most device APIs. Hybrid web applications are currently being developed using web widget engines such as provided by the BONDI/WAC [4] initiatives, device-independent frameworks such as the PhoneGap [5] application wrapper, and even completely web-centric operating systems such as Chromium OS [6] and HP webOS [7].

Current hybrid web application solutions, however, only

 ISBN: 978-1-61208-219-6

partially succeed in enabling a convincing ubiquitous experience [8]. Their main focus lies with porting traditional API support and operating system aspects to the web. Applications built upon these old principles result in virtual silos, unable to truly cross the physical boundaries of a device. By neglecting the evolution towards, e.g., distributed user interfaces, adaptive context-aware application behavior, etc. the true immersive nature of ubiquitous computing is mostly left behind [9] [10]. The absence of elaborate context-awareness is a key element driving this issue. In order for ubiquitous applications to adaptively support various contextual situations, the underlying application platform needs to provide structured, as well as secure and up-to-date access to the user's contextual setting. This requirement is not just limited to providing access to a detailed description of the target delivery context (screen size, interaction methods, available sensors, etc). Moreover, structured access to details regarding the user context (personal preferences, social context, disabilities, etc.) and the physical environment (location, time, etc.) ought to be supported.

From this perspective, we introduce the Webinos approach, a platform aiming to support hybrid web applications across mobile, PC, home media and in-car devices. Structured along a federated hierarchy, the proposed architecture enables developers to access a common set of rich context-aware APIs, allowing applications to dynamically adapt their cross-user, cross-service, and cross-device functionality in an open yet secure manner.

The remainder of this paper is structured as follows. Section II discusses background and related work. Section III provides a general overview of the proposed federated application platform. Section IV elaborates on the platform details of setting up secure context-awareness support. Section V discusses the use case of an adaptive social networking application. Finally, the conclusion and future work is outlined in Section VI.

II. Background and Related Work

The availability of detailed and reliable metadata regarding a user's contextual situation provides an important driver for enabling rich ubiquitous applications. The exact entities represented by this contextual information can be of a very dynamic nature, potentially affecting the consumer's expectations towards the application's user interface, behavior, content, etc. In initial context-aware research, context of use was considered a component containing only two parameters: the end-user's location and the set of objects in the immediate vicinity [11]. The subsequent introduction of extensible contextual categories has drastically increased the flexibility of this definition. Chen and Kotz hereto identified five contextual base categories: the device context, the user context, the environment context, the time context, and the historical context [12].

The device context describes the characteristics of the target device that is being used to access the application. A ubiquitous ecosystem covers a diversity of screen sizes, interaction methods, software support, etc. In web-based environments, the device capabilities are generally retrieved through Resource Description Framework (RDF) devices profiles, i.e., User Agent Profile (UAProf) [13] and Composite Capability/Preference Profiles (CC/PP) [14]. The necessary device identification step in this process is handled through HTTP header user agent matching. In order to facilitate the collection and aggregation of these device profiles, the W3C Mobile Web Initiative (MWI) standardized the Device Description Repository specification (DDR). The specification provides an API and its associated vocabulary for structured access to context providers services [15]. In essence, a DDR thus provides a standardized means for retrieving contextual information about a-priori knowledge on the characteristics of a particular target device or web runtime. Various open as well as proprietary DDR implementations are actively being maintained. Most notably OpenDDR, WURFL, and DeviceAtlas.

In a ubiquitous setting, the end-user's profile description gains more and more importance. Besides exposing information on user preferences and specific experience, this model should also comprise knowledge regarding the user's specific abilities and disabilities, e.g., enabling accessibility requirements for providing support to elderly people, and people with disabilities. From this perspective, Heckmann proposed the GUMO formalism as a general user model ontology for representing generic user descriptions using the Web Ontology Language semantics (OWL) [16]. The current challenge in this domain is modeling the enormous amount of parameters and relationships that characterize the user context [17]. To overcome this issue, forces are being joined with other ontology-driven projects such as Linked Data [18], and UbisWorld [19].

The environment-, time-, and historical context aspects define where, how, and when the interaction between the user and an application is exactly taking place. The environment context is specified by observing the numerous sensors available on the user's device (e.g., location, temperatures, network service discovery, the level of background noise, etc.). Furthermore, the notion of time and historical context is not to be neglected. As context is a dynamic concept, support for temporal patterns recognition and management is needed. The W3C Ubiquitous Web Domain is currently in the process of standardizing the Delivery Context Ontology specification (DCO) [20]. The DCO provides a formal model of the characteristics of the environment in which devices, applications, and services are operating.

III. Webinos Hybrid Application Platform

In order to enable application developers to set up services that fade out the physical boundaries of a device, we

 ISBN: 978-1-61208-219-6

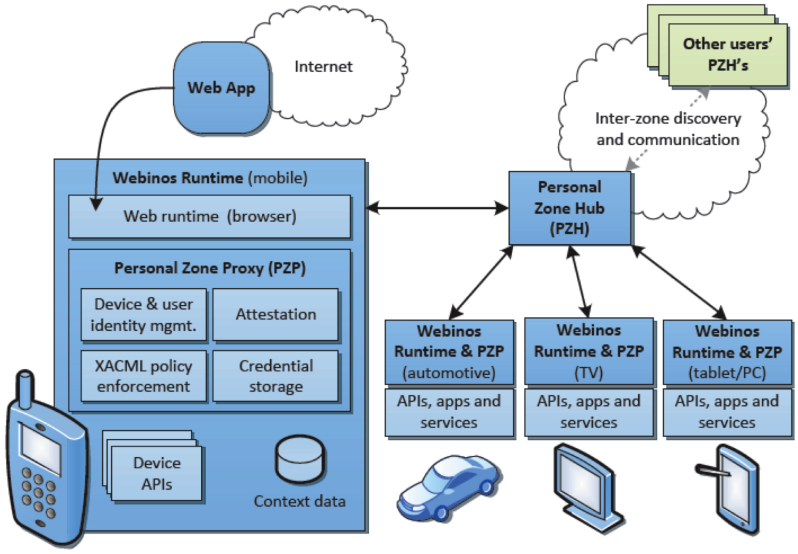


Figure 1. High-level Webinos platform overview

propose the Webinos architecture. Webinos is a federated web application platform and its runtime components are distributed over the devices, as well as the cloud. Figure 1 depicts a high-level overview of the platform's structure and deployment. The system's seamless interconnection principle is cornered around the notion of a so called Personal Zone. The Personal Zone represents a secure overlay network, virtually grouping a user's personal devices and services. To enable external access to and from the devices and services in this zone, the Webinos platform defines centralized Personal Zone Hub (PZH) components. Each user has his own PZH instance running in the cloud. The PZH is a key element in this architecture, as it contains a centralized repository of all contextual data in the Personal Zone. Moreover, the PZH keeps track of all devices and services in the zone and provides functionality to enable their mutual communication. This way, the PZH facilitates cross-device interaction with someone's services over the Internet. The PZHs are federated, allowing applications to easily discover and share data and services.

On the device-side, a Personal Zone Proxy (PZP) component is deployed. The PZP handles the direct communication with the zone's PZH. In order to keep the user's Personal Zone synchronized, the PZP is responsible for communicating device status with its PZP. This communication channel is built around a publisher-subscriber pattern [21]. As all external communication goes through the PZP, this component also acts as a policy enforcement point by managing all access to the device's exposed resources. In addition, the PZP is a fundamental component in upholding the Webinos platform's offline usage support. Although the proposed platform is designed with a strong focus on taking benefit from online usage, all devices in the Personal Zone have access to a locally synchronized copy of the data maintained by the PZH. The PZP can thus act in place of the PZH in case no reliable Internet connection can be established. This allows users to still operate the basic functionality of their applications even while being offline. All data to and from the PZP is again synchronized with the PZH as soon as the Internet access gets restored.

The Web Runtime (WRT) represents the last main component in the Webinos architecture. The WRT can be considered as the extension of a traditional web browser engine (e.g., WebKit, Mozilla Gecko). The WRT contains all necessary components for running and rendering web applications specified using standardized web technologies: HTML parser, JavaScript engine, CSS processor, rendering engine, etc. Furthermore, the WRT maintains a tight binding with the local PZP. The WRT-PZP binding allows the WRT to be much more powerful than traditional browser-based application environments. Through this binding, applications running in the WRT are able to securely interface with local device APIs and services. In addition, the PZP also allows the runtime to connect and synchronize with other devices in the Personal Zone through its binding with the PZH.

IV. Secure Context-aware Personal Zone

The innovative nature of the proposed approach lies with the platform's capability to establish a cross-device, cross-service, cross-user overlay network. For this Personal Zone concept to be successfully adopted by ubiquitous application developers, the platform needs to provide these developer access to a rich at-runtime overview of the user's contextual setting. As stipulated in Section I, elaborate platform support for transparent context management is vital. In this section, we provide more detail on the available developer tools for

 ISBN: 978-1-61208-219-6

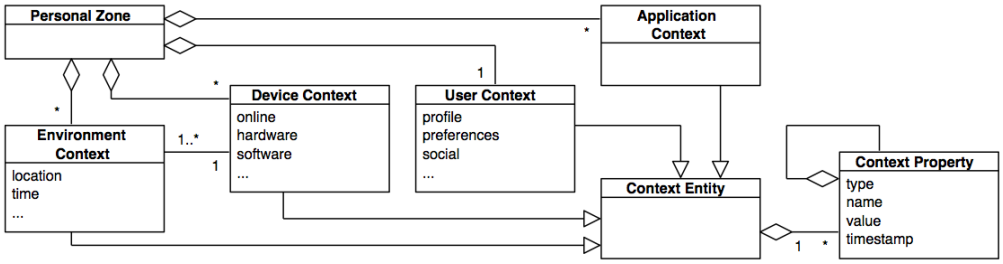


Figure 2. Simplified Webinos Personal Zone context model

setting up secure context-awareness within a Personal Zone environment.

A. Delivery Context Model

The Webinos delivery context model is defined to span all the platform's contextual knowledge within the user's Personal Zone. The model builds upon the W3C's Delivery Context Ontology (DCO) specification [20]. The Webinos delivery context model comprises four top-level submodels: the user context, the device context, the environment context, and the application context (see Figure 2 for a high-level overview). The first three submodels are internally managed and updated by the Webinos platform, whilst the application context model is to be maintained by the application developer. In order for each of these proposed models to support historical evaluation, pattern detection, conflict resolution strategies, all stored context properties are timestamped. The contextual information regarding the Personal Zone's owner is described by the user context model. This model consists of an aggregation of user profile data, user preferences, social context information, etc. Furthermore, each device and its physical environment are described by a separate instance of respectively the device context model and the environment context model. A device context model comprises knowledge regarding the corresponding device's availability in the Personal Zone, hardware characteristics, supported software, etc. The environment context model contains a description of a certain device's location, surrounding noise levels, etc. Lastly, the application context model provides developers the freedom to store a number of contextual properties, describing a situation from the perspective of their application.

B. Context Framework and API

The Webinos context framework is built on top of the above described context models. As depicted in Figure 1, providing application developers access to an elaborate distributed context framework is one of the core Webinos service. The Webinos context framework provides all necessary functionality for acquiring, storing, inferring new knowledge, and granting external access to the available contextualized data. Web applications running in the Webinos WRT, as well as other Webinos services, can rely on this framework to support their at-runtime need for contextualized data.

The Webinos Personal Zone is structured as a distributed system. In order to keep the zone synchronized, strong communication facilities between the device PZPs and the centralized PZH have architecturally been put in place. The Webinos context framework tries to makes optimal use of this structured communication channel to gain additional contextual knowledge regarding the Personal Zone. The context framework hooks into the PZP's event dispatching and synchronization mechanism. As visualized in Figure 3, out-bound status events are intercepted by the framework's context acquisition component and subsequently filtered for relevant data. The extracted context is locally stored and synchronized with the rest of the zone through a context-update event over the communication channel. The context acquisition process is autonomously managed by the Webinos platform and operates completely transparent for both the user and application developers. Moreover, the context framework is closely coupled with the PZP's security and policy enforcement framework. This binding ensures the secure handling of all context data that is being stored and accessed, as it often contains highly sensitive information.

For application developers aiming to create context-aware ubiquitous applications, the context framework provides an API to access Personal Zone wide context information. The context API supports the W3C standardized SPARQL RDF query language for unambiguously stating powerful context queries [22]. All context API requests are passed to the query processor component. The processor parses the request and checks its execution rights in collaboration with the PZP's policy enforcement framework. In case the request is granted by the PZP, the query is optimized and dispatched for execution. The API supports two modes for accessing context information: a generic query mode, and a change subscription mode. The generic query mode allows applications to execute targeted queries for specific context data in the storage system. The change subscription mode, on the other hand, enables an application to subscribe for specific context update events. These events are triggered by the context framework when new contextual knowledge is acquired.

 ISBN: 978-1-61208-219-6

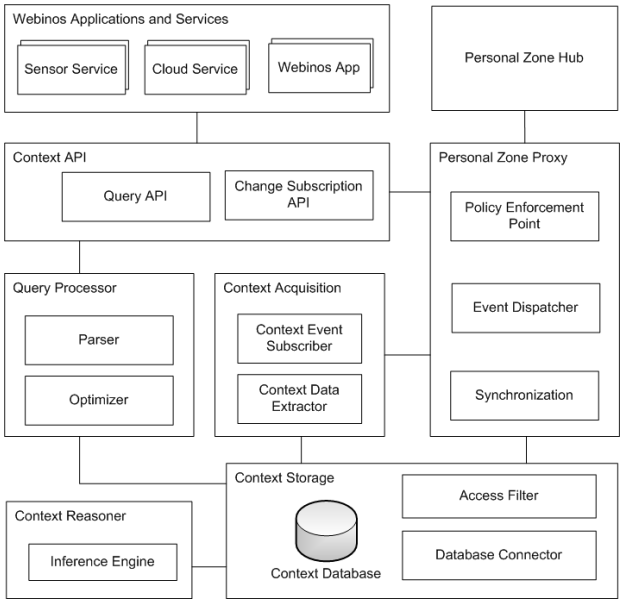


Figure 3. Webinos platform's distributed context framework, enabled through its tight integration with the Personal Zone

C. Policy Enforcement Support

The Webinos platform aims to meet the security and privacy requirements of applications and end users primarily through an access control policy system. Every access to a Webinos API is mediated by policies, enforced by the Personal Zone Proxies on each device as well as in the Personal Zone Hub. This action follows the principle of least privilege, granting applications only the permissions they require. Access policies are set when an application is first installed and can be updated subsequently. The policy system is derived from the BONDI/WAC architecture [4] and uses XACML (eXtensible Access Control Markup Language) including a number of extensions developed by the PrimeLife project [23]. XACML is a general-purpose access control language for defining policies based on subjects, resources, action and conditions [24]. By including the PrimeLife XACML extensions, the Webinos policy enhancement framework can allow users to specify detailed situation-specific access control policies. This is a significant advantage over current web runtime solutions and native mobile application platforms, where once an application has been granted access to a particular asset this access can be reused without further control.

Context data is often privacy-sensitive, as its analysis might reveal a user's history of actions or the people and devices they have interacted with. The Webinos platform aims to follow the principle of least surprise, so that a minimum of unexpected data disclosures will occur. This is achieved by disabling the collection of most context data by default, and providing the user a simple interfaces to turn it on again, complete with feedback about the kind of data that is being shared and stored. Where possible, data is filtered to remove unnecessary personal data. The main advantage of the Webinos platform is that context data remains within the zone and under the control of the end-user. This compares favorably to online user tracking, as users are able to view and manage the data stored about them, and applications will have to request specific access to this information.

V. Use Case

To elaborate on the application possibilities of the Webinos approach, we present a use case that has been built with the platform. The application is a cross-device social media app, able to use the APIs of television sets, mobile devices and desktop computers within a Personal Zone. The application utilizes the platform's default knowledge of a user's devices as well as their exposed capabilities and services. A user has the possibility to set policies for dispatching system API calls of the application to alternative devices. In result, the input (i.e., multimedia access, text input modality, contacts retrieval, etc.) and output (i.e., display selection) operations are adaptively abstracted from the traditional physical device level to the Personal Zone level. E.g., if a user wants to post a new message to one of his contacts on the Twitter social network: he can use his television set for displaying the main UI, use his smartphone an interaction device for navigating through the interface, putting in text, and accessing the device's contact list, access this home media center to attach a video to his post, etc.

A prototype of the proposed platform and use case are implemented and made available as part of the Webinos open source project [25]. Based on the project's extensive analysis of the current ubiquitous ecosystem [26], the following prototype platforms have been selected: PC (Linux, Windows, MacOS), mobile (Android), vehicles (Linux), home entertainment (Linux).

VI. Conclusion and Future Work

In this paper we presented the Webinos application platform approach, aiming to enable immersive ubiquitous software applications by leveraging the cross-platform possibilities of the web. The proposed approach utilizes the web infrastructure to establish its Personal Zone concept, a virtual overlay network for grouping all of user's devices and available services. Through the federated structure of Personal Zones, Webinos is able to provide application developers access to elaborate at-runtime context data regarding the current user, his devices, and the surrounding environment. The availability of this information allows developers to more accurately anticipate to a user's contextual situation. The Webinos platform's context-awareness enables numerous applications that make full use of the diversity and interconnectivity of devices. From this perspective, Webinos aims to be a key enabler in the realization of ubiquitous

 ISBN: 978-1-61208-219-6

applications that are able to execute across the physical boundaries of devices.

While the extensive evaluation of our approach has yet to be carried out, initial testing of prototype implementations shows promising results. Although the proposed platform addresses challenging issues in the ubiquitous application development domain, the current architecture only represents a first milestone in the pursuit of true ubiquitous application convergence. Whilst the Webinos platform provides structured access to rich contextual knowledge, it is still the application developers' responsibility to incorporate the necessary logic that allows their applications to act accordingly. Therefore, future work should include research on further extending the platform with (semi-) automated application adaptation mechanisms, driven by the platform's rich context-awareness. Regarding the privacy and security impact of such an application runtime, there will undoubtedly be a need to further experiment with user interfaces. This in order to strike an acceptable balance between the advantages that context sensitivity can offer, as well as privacy and user and developer convenience.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7-ICT-2009-5, Objective 1.2) under grant agreement number 257103 (Webinos project).

REFERENCES

[1] M. Tuters and K. Varnelis, "Beyond locative Media: giving Shape to the Internet of Things," *Leonardo*, vol. 39, no. 4, 2006, pp. 357-363.

[2] G. Banavar and A. Bernstein, "Challenges in design and software infrastructure for ubiquitous computing applications," *Advances in computers*, vol. 62, 2004, pp. 179-202.

[3] G. Lawton, "Moving the OS to the Web," *Computer*, vol. 41, no. 3, 2008, pp. 16-19.

[4] *Widget runtime high level technical specifications*, Wholesale Application Community (WAC) specification version 1, 2010.

[5] A. Charland and B. Leroux, "Mobile application development: web vs. native," *Communications of the ACM*, vol. 54, no. , 2011, pp. 49-53.

[6] W.T. Tsai, Q. Shao, X. Sun, and J. Elston, "Real-Time Service-Oriented Cloud Computing," in *Proc. IEEE 6th World Congr. on Services*, 2010, pp. 473-478.

[7] A. Weiss, "WebOS: say goodbye to desktop applications," *Networker*, vol. 9, no. 4, 2005, pp. 18-26.

[8] A. Taivalsaari and T. Mikkonen, "The Web as an Application Platform: The Saga Continues," in *Proc. IEEE 37th Conf. on Software Engineering and Advanced Applications*, 2011, pp. 170-174.

[9] H. Desruelle, D. Blomme, and F. Gielen, "Adaptive user interface support for ubiquitous computing environments," in *Proc. 2nd Int. Workshop on User Interface eXtensible Markup Language*, 2011, pp. 107-113.

[10] H. Desruelle, D. Blomme, and F. Gielen, "Adaptive mobile web applications through fine-grained progressive enhancement," in *Proc. 3rd Int. Conf. on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE)*, 2011, pp. 51-56.

[11] B. Schilit, N. Adams, and R. Want, "Context-aware Computing Applications," in *Proc. IEEE 1st Int. Workshop on Mobile Computing Systems and Applications*, 1994, pp. 85-90.

[12] G. Chen and D. Kotz, "A Survey of Context-aware Mobile Computing Research," Dept. Computer Science, Dartmouth College, Tech. Rep. TR2000-381, 2000.

[13] *WAG UAProf*, Wireless Application Protocol specification WAP-248-UAPROF-2001020, 2001.

[14] *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*, W3C recommendation, 2004.

[15] *Device Description Repository Core Vocabulary*, W3C working group note, 2008.

[16] D. Heckmann, "Ubiquitous User Modeling," Ph.D. dissertation, Dept. of Computer Science, Saarland University, 2005.

[17] J.L.T. Silva, A. Moreto Ribeiro, E. Boff, T. Primo, and R.M. Viccari, "A Reference Ontology for Profile Representation in Communities of Practice," *Metadata and Semantic Research*, vol. 240, 2011, pp. 68-79.

[18] T. Heath and C. Bizer, "Linked data: Evolving the web into a global data space," in *Synthesis Lectures on the Semantic Web: Theory and Technology*, vol. 1, no. 1. Morgan & Claypool, 2011, pp. 1-136.

[19] D. Heckmann, M. Loskyll, R. Math, P. Recktenwald, and C. Stahl, "Ubisworld 3.0: A Semantic Tool Set for Ubiquitous User Modeling," in *Proc. 17th Int. Conf. on User Modeling, Adaptation and Personalization*, 2009.

[20] *Delivery Context Ontology*, W3C working group note, 2010.

[21] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-oriented software architecture: A system of patterns*. West Sussex: John Wiley & Sons, 2001.

[22] T. Segaran, C. Evans, and J. Taylor, *Programming the Semantic Web*. Sebastropol: O'Reilly Media, 2009.

[23] C.A. Ardagna, S. De Capitani Di Vimercati, E. Pedrini, and P. Samarati, "Primelife policy language," in *Proc. W3C workshop on access control application scenarios*, 2009.

[24] *eXtensible Access Control Markup Language (XACML) version 1.1*, OASIS standard, 2003.

[25] Webinos, "Webinos Developer Portal," Available: https://developer.webinos.org/ [May. 1, 2012].

[26] Webinos, "Industry landscape, governance, licensing and IPR frameworks," Tech. Rep. D02.3, 2011.

 ISBN: 978-1-61208-219-6

Adaptive Fractal-like Network Structure for Efficient Search of Targets at Unknown Positions

Yukio Hayashi
Graduate School of Knowledge Science
Japan Advanced Institute of Science and Technology
Nomi-city, Ishikawa-pref., Japan
Email: yhayashi@jaist.ac.jp

***Abstract*—Since a spatial distribution of communication requests is inhomogeneous and related to a population, in constructing a network, it is crucial for delivering packets on short paths through the links between proximity nodes and for distributing the load of nodes how to locate the nodes as base-stations on a realistic wireless environment. In this paper, from viewpoints of complex network science and biological foraging, we propose a scalably self-organized geographical network, in which the proper positions of nodes and the network topology are simultaneously determined according to the population, by iterative divisions of rectangles for load balancing of nodes in the adaptive change of their territories. In particular, we consider a decentralized routing by using only local information, and show that, for searching targets around high population areas, the routing on the naturally embedded fractal-like structure by population has higher efficiency than the conventionally optimal strategy on a square lattice.**

***Keywords*-self-organised design; wide-area wireless communication; routing in ad hoc networks; random walk; Levy flight.**

I. INTRODUCTION

Many network infrastructures: power grids, airline networks, and the Internet, are embedded in a metric space, and long-range links are relatively restricted [1], [2] for economical reasons. The spatial distribution of nodes is neither uniformly at random nor on a regular lattice, which is often assumed in the conventional network models. In real data, a population density is mapped to the number of router nodes on Earth [1]. Similar spatially inhomogeneous distributions of nodes are found in air transportation networks [3] and in mobile communication networks [4]. Thus, it is not trivial how to locate nodes on a space in a pattern formation of points. Point processes in spatial statistics [5] provide models for irregular patterns of points in urban planing, astronomy, forestry, or ecology, such as spatial distributions of rainfall, germinations, plants, and animals. The processes assume homogeneous Poisson and Gibbs distributions to generate a pattern of random packing or independent clustering, and to estimate parameters of competitive potential functions in a territory model for a given statistical data, respectively. However, rather than random pattern and statistical estimation, we focus on a self-organized network infrastructure by taking into account realistic spatial distributions of nodes and communication requests. In particular, we aim to develop adaptive and scalable ad hoc networks by adding the links between proximity nodes according to the increasing of communication requests. Because a spatial distribution of communication requests affect the proper positions of nodes, which control both the load of requests assigned to each node (e.g., assigned at the nearest access point of node as a base-station from a user) and the communication efficiency depending on the selection of routing paths.

For a routing in ad hoc networks, global information, e.g., a routing table in the Internet, cannot be applied, because many nodes and connections between them are likely to change over time. In early works on computer science, some decentralized routing methods were developed to reduce energy consumption in sensor or mobile networks. However they lead to the failure of guaranteed delivery [6]; in the flooding algorithm, multiple redundant copies of a message are sent and cause congestion, while greedy and compass routings may occasionally fall into infinite loops or into a dead end. In complex network science, other efficient decentralized routing methods have been also proposed. The stochastic methods by using local information of the node degrees and other measures are called preferential [7] and congestion-aware [8], [9] routings as extensions of a uniformly random walk.

Decentralized routing has a potential performance to search a target whose position is unknown in advance. Since this situation looks like foraging, the biological strategy may be useful for the efficient search. We are interested in a relation of the search and the routing on a spatially inhomogeneous network structure according to a population. Many experimental observations for biological foraging found the evidence in favor of anomalous diffusion in the movement of insects, fishes, birds, mammals and human being [10]. As the consistent result, it has been theoretically analyzed for a continuous space model that an inverse square root distribution of flight lengths is an optimal strategy to search sparsely and randomly located targets on a homegeneous space [11]. The discrete space models on a regular lattice

 ISBN: 978-1-61208-219-6

[12] and the defective one [13] are also discussed. Such behavior is called *Levy flight* characterized by a distribution function $P(l_{ij}) \sim l_{ij}^{-\mu}$ with $1 < \mu \leq 3$, where l_{ij} is a flight length between nodes i and j in the stochastic movement for any direction. The values of $\mu \geq 3$ lead to Brownian motions, while $\mu \to 1$ to ballistic motions. The optimal case is $\mu \approx 2$ for maximizing the efficiency of search. Here, we assume that the mobility of a node is ignored due to a sufficiently slow speed in comparison with the communication process. In the current or future technologies, wide-area wireless connections by directional beams will be possible, the modeling of unit disk graph with a constant transmission range is not necessary. Thus, we propose a scalably self-organized geographical network and show that the naturally embedded fractal-like structure is suitable for searching inhomogeneously distributed targets more efficiently than the square lattice tracked by the Levy flights.

II. Geographical Networks

We introduce geographical network models proposed in complex network science.

A. Conventional Models

Geographical constructions of complex networks have been proposed so far. As a typical generation mechanism of scale-free (SF) networks that follow a power law degree distribution found in many real systems [14], [15], a spatially preferential attachment is applied in some extensions [16], [17], [18], [19], [20] from the topological degree based model [21] to a combination of degree and distance based model. On the other hand, geometric construction methods have also been proposed. They have both small-world [22] and SF structures generated by a recursive growing rule for the division of a chosen triangle [23], [24], [25], [26] or for the attachment aiming at a chosen edge [27], [28], [29] in random or hierarchical selections. These models are proper for the analysis of degree distribution due to the regularly recursive generation process. Although the position of a newly added node is basically free as far as the geometric operations are possible, it has no relation to population. Considering the effects of population on a geographical network is necessary to self-organize a spatial distribution of nodes that is suitable for socioeconomic communication and transportation requests. Moreover, in these geometric methods, narrow triangles with long links tend to be constructed, and adding only one node per step may lead to exclude other topologies from the SF structure. Unfortunately, SF networks are extremely vulnerable against the intentional hub attacks [30]. We should develop other self-organizations of network apart from the conventional models; e.g., a better network without long links can be constructed by subdivisions of equilateral triangle, which is a well balanced (neither fat nor thin) shape for any directions.

B. Generalized Multi-Scale Quartered Network

Thus, we have considered the multi-scale quartered (MSQ) network model [31], [32]. It is based on a stochastic construction by a self-similar tiling of primitive shape, such as an equilateral triangle or square. The MSQ networks have several advantages of the strong robustness of connectivity against node removals by random failures and intentional attacks, the bounded short path as $t = 2$-spanner [33], and the efficient face routing by using only local information. Furthermore, the MSQ networks are more efficient (economic) with shorter link lengths and more suitable (tolerant) with lower load for avoiding traffic congestion [32] than the state-of-the-art geometric growing networks [23], [24], [25], [26], [27], [28], [29] and the spatially preferential attachment models [16], [17], [18], [19], [20] with various topologies ranging from river to SF geographical networks. However, in the MSQ networks, the position of a new node is restricted on the half-point of an edge of the chosen face, and the link length is proportional to $(\frac{1}{2})^H$ where H is the depth number of iterative divisions. Thus, from square to rectangle, we generalize the division procedures as follows.

- Step0: Set an initial square whose inside are the candidates of division axes as the segments of a $L \times L$ lattice.
- Step1: At each time step, a face is chosen with a probability proportional to the population counted in the face covered by mesh blocks of a census data.
- Step2: Four smaller rectangles are created from the division of the chosen rectangle face by horizontal and vertical axes. For the division, two axes are chosen by that their cross point is the nearest to the population barycenter of the face.
- Step3: Return to Step 1, while the network size (the total number of nodes) N does not exceed a given size.

Note that the maximum size N_{max} depends on the value of L; the iteration of division is finitely stopped, since the extreme rectangle can not be divided any longer when one of the edge lengths of rectangle is the initial lattice's unit length. We use the population data on a map in $80km^2$ of 160×160 mesh blocks ($L = 160$) provided by the Japan Statistical Association. Of course, other data is possible.

It is worth noting that the positions of nodes and the network topology are simultaneously determined by the divisions of faces within the fractal-like structure. There exists a mixture of sparse and dense parts of nodes with small and large faces. Moreover, with the growing network, the divisions of faces perform a load balancing of nodes in their adaptively changed territories for the population. We emphasize that such a network is constructed according to a spatially inhomogeneous distribution of population, which is proportional to communication requests in a realistic environment. In the following, we show the naturally embedded fractal-like structure are suitable for searching targets.

 ISBN: 978-1-61208-219-6

III. Search Performance

As a preliminary, we consider the preferential routing [7] which is also called α-random walk [34]; The forwarding node j is chosen proportionally to K_j^α by a walker in the connected one hop neighbors $\mathcal{N}_i$ of its resident node i of a walker (packet), where K_j denotes the degree of node j and α is a real parameter. We assume that the start position of walker is set to the nearest node to the population barycenter of the initial square. Figure 1 shows the length distribution of visited links. The dashed lines in log-log plot suggest a power law, for which the exponents estimated as the slopes by a mean-square-error method are 2.336, 2.315, and 2.296 for $\alpha = 1, 0, -1$, respectively. These values are close to the optimal exponent $\mu \approx 2$ [11], [12] in the Levy flight on a square lattice. The exponents for the α-random walks slightly increase as the network size N becomes larger. Here, the case of $\alpha = 0$ shows the length distribution of existing links on a network. Since the stationary probability of incoming at node j is $P_j^\infty \propto K_j^{1+\alpha}$ [35], especially at $\alpha = 0$, each of the connected links to j is chosen at random by the probability $1/K_j$ for the leaving from j, therefore a walker visit each link at the same number. Figure 2 shows that the frequency of visited links by the α-random walks at $\alpha = \pm 1$ is different even for the degrees 3 and 4 in a generalized MSQ network. On the thick lines, a walker tends to visit high population (diagonal) areas colored by orange and red in the case of $\alpha = 1$, while it tends to visit low population peripheral (corner) areas in the case of $\alpha = -1$. Thus, the case of $\alpha = 1$ is expected to selectively cover high population areas, which has a lot of communication requests in cities. Note that the absolute value of α should be not too large, since a walker is trapped between high/low degree nodes in a long time as the ping-pong phenomena that does not contribute to the search of targets.

We investigate the search efficiency for the α-random walk on a generalized MSQ network, and compare the efficiency with that for Levy flights on a $L \times L$ square lattice with periodic boundary conditions [12]. As shown in Fig. 3, a walker constantly looks for targets (destination nodes of packets) scanning on a link between two nodes in the generalized MSQ network. If a target exists in the vision area of r_v hops for the up/down/left/right directions from the center position, a walker gets it and return to the position on the link for continuing the search on the same direction. When more than one targets exit in the area, a walker gets all of them successively in each direction, and return to the position. Only at a node of rectangle, the search direction is changeable along one of the connected links. Thus, the search is restricted on the edges of rectangle in the generalized MSQ network. While the search direction of a Levy flight on the square lattice [12] is selectable from four directions of horizontal and vertical at all times after getting a target in the scanning with the vision area of r_v

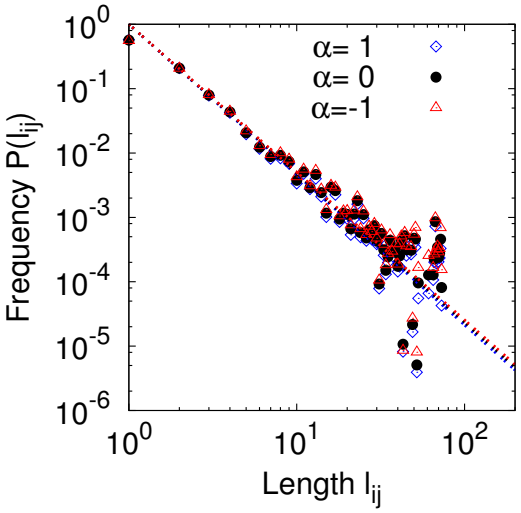


Figure 1. Length distribution of visited links on generalized MSQ networks by an α-random walker in 10^6 time steps. The marks of blue diamond, black circle, and red triangle correspond to the cases of $\alpha = 1, 0, -1$, respectively. These results are obtained by the average of 100 networks for $N = 2000$.

hops, moreover, the length of scan follows $P(l_{ij}) \sim l_{ij}^{-\mu}$, $l_{ij} > r_v$. We set a target at the position chosen proportionally to the population around a cross point in $(L+1)^2$, for which the population is defined by the average of four values in its contact mesh regions. In particular, we discuss the destructive case [12]: once a target is detected by a walker, then it is removed and a new target is created at a different position chosen with the above probability.

The search efficiency [11], [12], [13] is defined by

$$\eta \stackrel{\text{def}}{=} \frac{1}{M} \sum_{m=1}^{M} \frac{N_s}{L_m}, \tag{1}$$

$$\lambda \stackrel{\text{def}}{=} \frac{(L+1)^2}{N_t 2 r_v}, \tag{2}$$

where L_m denotes the traversed distance counted by the lattice's unit length until detecting $N_s = 50$ targets from the total N_t targets in the mth run. We consider a variety of $N_t = 60, 100, 200, 300, 400$, and 500 for investigating the dependency of the search efficiency on the number N_t of targets. The quantity λ represents the mean interval between two targets for the scaling of efficiency by target density. We set $M = 10^3$ and $r_v = 1$ for the convenience of simulation. Intuitively, the sparse and dense structures according to the network size N have the advantage and disadvantage in order to raise the search efficiency in the generalized MSQ network. Although the scanned areas are limited by some large rectangle holes as N is small, a walker preferably visits the high population areas that include many targets. While the scanned areas are densely covered as N is large, the

 ISBN: 978-1-61208-219-6

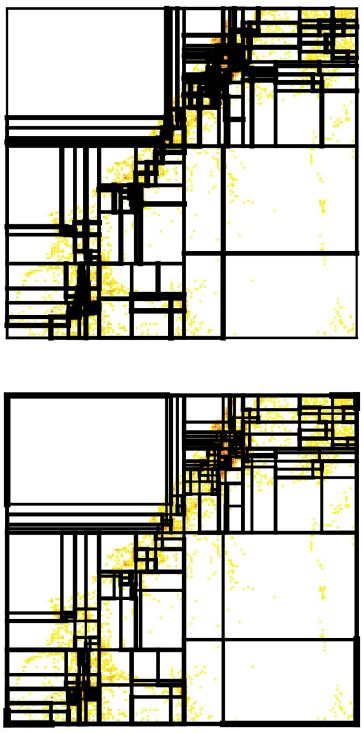


Figure 2. Visualization examples of the visited links by α-random walks at $\alpha = 1$ (Top) and $\alpha = -1$ (Bottom) on a generalized MSQ network for $N = 500$. The thickness of link indicates the frequency of visiting in 10^6 time steps. From light to dark: white, yellow, and orange to red, the color gradation on a mesh block is proportionally assigned to the population. Many nodes represented as cross points of links concentrate on high population (dark: orange and red) areas on the diagonal direction. In the upper left and lower right of square, corner triangle areas lighted by almost white are the sea of Japan and the Hakusan mountain range.

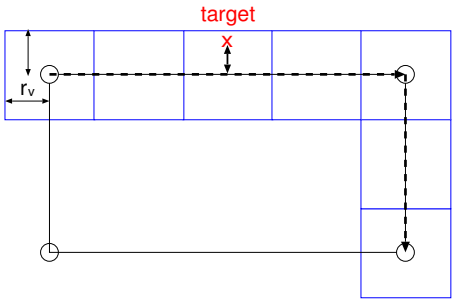


Figure 3. Searching in a generalized MSQ network. Each blue square represents a vision area, and is scanned (from left to right, from top to bottom in this example) by the walker on an edge between two nodes (denoted by circles) of a rectangle. For a target in the area, the walker moves to get it and returns on the link.

search direction is constrained on long links of a collapse rectangle, therefore it is rather hard for a walker to escape from a local area in which targets are a few.

We compare the search efficiency of α-random walks in the generalized MSQ networks with that of the Levy flights

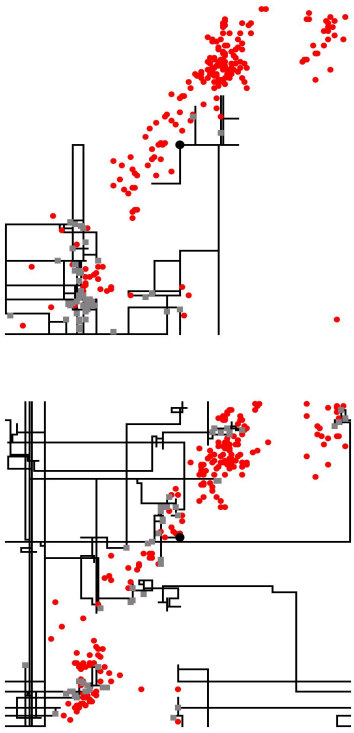
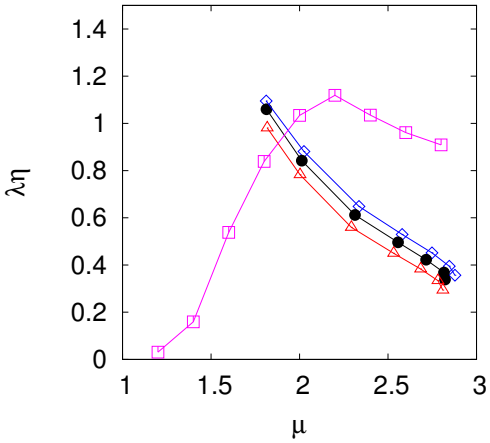


Figure 4. Trajectories of a random walk (Top) at $\alpha = 0$ on a generalized MSQ network for $N = 500$ and of a Levy flight (Bottom) for $\mu = 1.8$ on the square lattice with periodic boundary conditions until detecting $N_s = 50$ targets in $N_t = 200$. Black circle, red circles, and gray rectangle marks denote the start point at the population barycenter, the existing targets, and the removed targets after the detections, respectively. Note that a walker can travel back and forth on a link in the connected path.

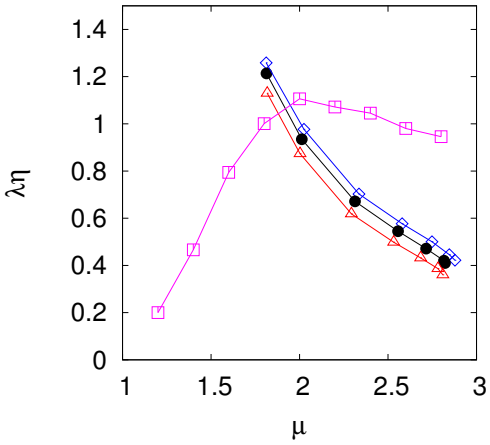
in the square lattice. Figure 4 shows typical trajectories until detecting $N_s = 50$ targets. On the generalized MSQ network and the square lattice, a walker tends to cover a local area with high population and a wider area, respectively. Without wandering peripheral wasteful areas, the generalized MSQ network has a more efficient structure than the square lattice for detecting many targets concentrated on the diagonal areas. Here the exponent $\mu = 1.8$ of Levy flight corresponds to the slope of $P(l_{ij})$ in the generalized MSQ network at the optimal size $N = 500$ for the search efficiency. As shown in Fig. 5(a)(b), the generalized MSQ networks of $N = 500$ (the diamond, circle, and triangle marks are sticking out at the left) have higher efficiency than the square lattice (the rectangle mark). For the cases with many nodes of $N \geq 1000$, the efficiency is decreased more rapidly than that of the Levy flight, however this phenomenon means that many nodes are wasteful and unnecessary to get a high search performance in generalized MSQ networks. When the number N_t of targets increases in cases from Fig. 5(a) to (b), the curves are shift up, especially for the generalized MSQ networks. The peak value for $N_t = 200$ is larger than the optimal case of the Levy flight at $\mu = 2.0$. Therefore

 ISBN: 978-1-61208-219-6

denser targets to that extent around $N_t = 200$ is suitable, although a case of larger $N_t > 300$ brings down the search efficiency even for inhomogeneously distributed targets.



(a) $N_t = 100$



(b) $N_t = 200$

Figure 5. The scaled efficiency $\lambda\eta$ vs. the exponent μ. The marks of blue diamond, black circle, and red triangle correspond to the cases of $\alpha = 1, 0, -1$, respectively, in which the increasing values of μ are estimated for generalized MSQ networks at $N = 500, 1000, 2000, 3000, 4000, 5000$, and 5649: N_{max} from left to right. The magenta rectangle corresponds to the case of Levy flights on the square lattice. These results are obtained by the average of 100 networks.

In more details, Fig. 6 shows the effect of the number N_t of targets on the search efficiency $\lambda\eta$. The efficiency firstly increases, then reaches at a peak, and finally decreases for setting more targets. This up-down phenomenon is caused from a trade-off between L_m and N_t in Eqs. (1) and (2). Note that the case of size $N < 500$ is omitted for the generalized MSQ networks. Because sometimes the process for detecting targets until N_s is not completed, moreover, the variety of link lengths is too little to estimate the exponent as a slope of $P(l_{ij})$ in the log-log scale. In other words, the estimation is inaccurate because of the short linear part.

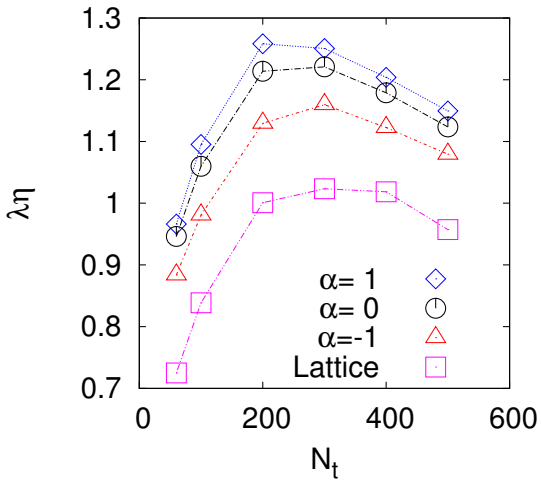


Figure 6. The number N_t of targets vs. the scaled efficiency $\lambda\eta$ of α-random walks on the generalized MSQ networks for $N = 500$ and of the corresponding Levy flights for $\mu = 1.8$ (see Fig. 5) on the square lattice. The maximum (optimal) efficiency appears in $N_t = 200 \sim 300$. These results are obtained by the average of 100 networks.

IV. Conclusion

We have considered a scalably self-organized geographical network by iterative divisions of rectangles for load balancing of nodes in the adaptive change of their territories according to the increasing of communication requests. In particular, the spatially inhomogeneous distributions of population and the corresponding positions of nodes are important. For the proposed networks, we have investigates the search efficiency in the destructive case [12] with new creations of target after the detections, and shown that the α-random walks as decentralized routing on the networks have higher search efficiency than the Levy flights known as the optimal strategy [11], [12] on the square lattice with periodic boundary conditions. One reason for the better performance is the anisotropic covering of high population areas. Thus, the naturally embedded fractal-like structure is suitable for searching targets in such a realistic situation. In more rigorous discussions about the performance, statistical tests [36] may be useful to clarify the applicability of the proposed method.

Acknowledgment

The author would like to thank Mitsugu Matsushita (Chuo University) for his valuable comments and Takayuki Komaki (JAIST) for helping the simulation. This research is supported in part by Grant-in-Aide for Scientific Research in Japan, No.21500072.

References

[1] S. H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the internet's large-scale topology," *PNAS*, 99(21), 13382–13386, 2002.

 ISBN: 978-1-61208-219-6

[2] M. T. Gastner, and M. E. J. Newman, "The spatial structure of networks," *Eur. Phys. J. B*, 49(2), 247–252, 2006.

[3] R. Guimerà, S. Mossa, A. Turtschi, and L. Amaral, "The worldwide air transportation network: Anormalous centrality, community structure, and cities' global roles," *PNAS*, 102(22), 7794–7799, 2005.

[4] R. Lambiotte, V. Blondel, C. de Kerchove, E. Huens, C. Prieur, Z. Smoreda, and P. Dooren, "Geographical dispersal of mobile communication networks," *Physica A*, 387, 5317–5325, 2008.

[5] D. Stoyan, W. Kendall, and J. Mecke, *Stochastic Geometry and its Applications (2nd ed.)*, John Wiley & Sons, 1995.

[6] J. Urrutia, "Routing with Guaranteed Delivery in Geometric and Wireless Networks," *in Handbook of Wireless Networks and Mobile Computing, I. Stojmenović(ed)*, Chapter 18, John Wiley & Sons, 2002.

[7] W.-X. Wang, B.-H. Wang, C.-Y. Yin, Y.-B. Xie, and T. Zhou, Traffic dynamics based on local routing protocol on a scale-free network. *Phys. Rev. E*, 73, 026111, 2006.

[8] B. Danila, Y. Yu, S. Earl, J. A. Marsh, Z. Toroczkai, and K. E. Bassler, "Congestion-gradient driven transport on complex networks," *Phys. Rev. E*, 74, 046114, 2006.

[9] W.-X. Wang, C.-Y. Yin, C.-Y., G. Yan, and B.-H. Wang, "Integrating local static and dynamic information for routing traffic," *Phys. Rev. E*, 74, 016101, 2006.

[10] G. M. Viswanathan, M. G. E., Luz, E. P. Raposo, and H. E. Stanley, *The Physics of Foraging -An Introduction to Random Searches and Biological Encounters*, Cambridge University Press, 2011.

[11] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E., da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," *Nature*, 401, 911–914, 1999.

[12] M. C. Santos, G. M. Viswanathan, E. P. Raposo, and M. E. da Luz, "Optimization of random search on regular lattices," *Phys. Rev. E*, 72, 046143, 2005.

[13] M. C. Santos, G. M. Viswanathan, E. P. Raposo, and M. E. da Luz, "Optimization of random searches on defective lattice networks," *Phys. Rev. E*, 77, 041101, 2008.

[14] R. Albert, and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, 74(1), 47–97, 2002.

[15] N. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, 45(2), 167–256, 2003.

[16] R. Xulvi-Brunet, and I. Sokolov, "Evolving networks with disadvantaged long-range connections," *Phys. Rev. E*, 66, 026118, 2002.

[17] S. S.Manna, and P. Sen, "Modulated scale-free network in euclidean space," *Phys. Rev. E*, 66, 066114, 2002.

[18] P. Sen, and S. S. Manna, "Clustering properties of generalized critical euclidean network," *Phys. Rev. E*, 68, 026104, 2003.

[19] A. K. Nandi, and S. S. Manna, "A transition from river networks to scale-free networks," *New J. Phys.*, 9, 30, 2007

[20] J. Wang, and G. Provan, "Topological analysis of specific spatial complex networks," *Advances in Complex Systems*, 12(1), 45–71, 2009.

[21] A.-L. Barabási, and R. Albert, "Emergence of scaling in random networks," *Science*, 286, 509–512, 1999.

[22] D. J. Watts, and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, 393, 440–442, 1998.

[23] Z. Zhang, S. Zhou, Z. Su, T. Zou, and J. Guan, "Random sierpinski network with scale-free small-world and modular structure," *Euro. Phys. J. B*, 65, 141–148, 2008.

[24] T. Zhou, G. Yan, and B.-H. Wang, "Maximal planar networks with large clustering coefficient and power-law degree distribution," *Phys. Rev. E*, 71, 046141, 2005.

[25] Z. Zhang, and L. Rong, "High-dimensional random apollonian networks," *Physica A*, 364, 610–618, 2006.

[26] J. P. K. Doye, and C. P. Massen, "Self-similar disk packings as model spatial scale-free networks," *Phys. Rev. E*, 71, 016128, 2005.

[27] L. Wang, F. Du, H. P. Dai, and Y. X. Sun, "Random pseudofractal scale-free networks with small-world effect," *Eur. Phys. J. B*, 53, 361–366, 2006.

[28] H. D. Rozenfeld, S. Havlin, and D. ben Avraham, "Fractal and transfractal scale-free nets," *New J. of Phys.*, 9, 175, 2007.

[29] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, "Pseudofractal scale-free web," *Phys. Rev. E*, 65, 066122, 2002.

[30] R. Albert, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, 406, 378–382, 2000.

[31] Y. Hayashi, "Evolutionary construction of geographical networks with nearly optimal robustness and efficient routing properties," *Physica A*, 388, 991–998, 2009.

[32] Y. Hayashi, and Y. Ono, "Geographical networks stochastically constructed by a self-similar tiling according to population," *Phys. Rev. E*, 82, 016108, 2010.

[33] M. I. Karavelas, and L. J. Guibas, "Static and kinetic geometric spanners with applications," In *Proc. of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pp. 168–176, 2001.

[34] Y. Hayashi, and Y. Ono, "Traffic properties for stochastic routing on scale-free," *IEICE Trans. on Communication*, E94-B(5), 1311–1322, 2011.

[35] J. D. Noh, and H. Rieger, "Random walks on complex networks," *Phys. Rev. Lett.*, 92(11), 118701, 2004.

[36] A. Clauset, C. R. Shalizi, M .E. J. Newman, "Power-Law Distributions in Empirical Data," *SIAM Review*, 51, 661–703, 2009.

 ISBN: 978-1-61208-219-6

On the Adaptivity of Distributed Association Rule Mining Agents

Adewale Opeoluwa Ogunde
Dept. of Mathematical Sciences
Redeemer's University (RUN),
Redemption Camp, Nigeria
ogundea@run.edu.ng

Olusegun Folorunso
Dept. of Computer Science
Federal University of Agriculture,
Abeokuta, Nigeria
folorunsolusegun@yahoo.com

Adesina Simon Sodiya
Dept. of Computer Science
Federal University of Agriculture,
Abeokuta, Nigeria
sina_ronke@yahoo.co.uk

***Abstract*—Current and real association rule mining tasks can only be successfully done in a distributed setting where transaction data sites are mined dynamically and appropriately as they are updated. Mobile agents' paradigms are now used to mine association rules in such circumstances. As these mobile agents travel in the distributed association rules mining environment, they are liable to unforeseen changes, circumstances and faults that may arise in these environments. Few researches had been carried out on the adaptivity of mobile agents, but the adaptivity of distributed association rule mining agents is yet to be explored. Therefore, this work examines an adaptive architectural framework that mines association rules across multiple data sites, and more importantly the architecture adapts to changes in the updated database and the mining environment giving special considerations to the incremental database. This system was made adaptive both at the algorithm level and the mining agent level. Adaptation at the mobile agent level uses sensors to sense environmental changes, creates a percept of the environment and sends it to the adapter which adapts to the environmental changes by dynamically changing the goals of the mining agents or maintaining the original goals. The system promises to efficiently generate new and up-to-date rules while also adapting to faults and other unforeseen circumstances in the distributed association rules mining environment without the usual user's interference. The model presented here provided the background ideas needed for the development of adaptive distributed association rule mining agents.**

Keywords-adaptive agents; distributed association rule mining; distributed databases; knowledge integration; mobile agents

I. INTRODUCTION

Association rule mining (ARM) finds frequent patterns, associations, correlations, or casual structures sets of items or objects from large databases [1]. The idea is to find out the relation or dependency of occurrence of one item based on occurrence of other items. Distributed Association Rule Mining is the process of mining association rules and patterns from distributed data sources. Mobile agents are any relatively autonomous entity able to perform actions in an environment perceived by it. Mobile agents' paradigm [7] has several advantages among which are: conservation of bandwidth and reducing latencies while also complex, efficient and robust behaviors can be realized with surprisingly little code. Performance of these mining agents may be hampered in the distributed association rule environments due to faults and other unforeseen circumstances. Therefore, in this research, we capitalized on the power of agents to introduce an adaptive distributed association rule mining agents that mines across distributed databases while adapting to unforeseen changes in the entire system. The organization of the rest of this paper is as follows. Section II provides a review of some existing and related works. Section III describes the design details of the adaptive distributed association rule mining agents. Finally, section IV contains some concluding remarks and scope for future work.

II. LITERATURE REVIEW

This section reviews existing work on distributed association rule mining, agents and adaptive systems.

A. Association Rule Mining

Association Rule Mining (ARM) is one of the most popular tasks of Data Mining (DM). Data mining is a powerful new technology with great potential to help companies focus on the most important information in the data they have collected about the behavior of their customers and potential customers [2]. It finds patterns in data that show associations between domain elements. DM is generally focused on transactional data, such as a database of purchases at a store. This task is known as Association Rule Mining (ARM), and was first introduced in Agrawal et al. [1]. An association rule is of the form X → Y, where X and

 ISBN: 978-1-61208-219-6

Y are disjoint conjunctions of attribute-value pairs. The most commonly used mechanism for determining the relevance of identified ARs is the support and confidence framework. The confidence of the rule is the conditional probability of Y given X, that is Pr(Y|X). The support of the rule is the prior probability of X and Y, that is Pr(X and Y). Distributed association rule mining (DARM) refers to the mining of association rules from distributed data sets. The data sets are stored in local databases hosted by local computers which are connected through a computer network [3]. Typical DARM algorithms involve local data analysis from which a global knowledge can be extracted using knowledge integration techniques [4]. A review of current distributed association rule mining methods was presented by Ogunde et al. [5]. Albashiri [6] gave some key issues to be addressed for distributed data mining tasks dwelling so much on the extendability of the system but the adaptivity has so far not been addressed by researchers.

B. Agents and Multi-Agent Systems (MAS)

Agents are defined by Wooldridge [7] as computer software that are situated in some environment and are capable of autonomous action in this environment in order to meet their design objectives. Agents are active, task-oriented, modeled to perform specific tasks and capable of decision making. By combining multiple agents, in one system, to solve a problem, the resultant system is a MAS. From the literature, well documented advantages of MAS includes: Decentralized control, Robustness, Simple extendability, Sharing of expertise and Sharing of resources [7]. According to Wooldridge [8], the cognitive functions of a rational agent are categorized into the following three modalities. First, beliefs are facts which the agent holds, which represent the properties about the agent's environment. Ideally, the agent's current belief set should be consistent. Second is that desires are the agent's long term goals. There is no requirement that the agent's desires should be consistent. Third modality is that intentions represent a staging post between beliefs and desires, in that they represent goals or sub-goals that the agent intends to actually bring about.

C. Adaptivity in Multi-Agent Systems

Agents typically operate in dynamic environments. Agents come and go, objects appear and disappear, and cultures and conventions change. Whenever an environment of an agent changes to the extent that an agent is unable to cope with (part of) the environment, an agent needs to adapt. Changes in the social environment of an agent, for example, may require modifications to existing agents [9]. The ability to adapt to dynamic environment and unexpected events is a key issue for mobile agents [10]. These inherent changes are dynamic in nature and demands that multi-agent systems should be adaptive and flexible. Therefore, for a multi-agent system, adaptation represents the ability of the multi-agent system to recognize and response to unanticipated internal and external change. Few researches have been done on agents' adaptability but there are none on the adaptivity of DARM agents [9]. Most especially, adaptive agents proposed in this work were based on the foundation laid by Ranjan et al. [10] for adaptive mobile agents.

Lacey and Hexmoor [11] addressed the question of assigning social norms to agents that will eventually act in complex dynamic environments and also treated the possibility of allowing the agents to adapt to new situations as they arise, and choose their norms accordingly. The researchers argued that adaptation is preferable to prescription, in that agents should be allowed to revise their norms when there is a need to adapt to new situations by revising their norms as appropriate. They also argued that their approach is better than prescribing norm adherence at design time. In Lacey and Hexmoor [11], a system was constructed in which the performance of multiple agents operating in the same environment were assessed and experimental results showed that in some circumstances adaptive norm revision strategies performed better than prescriptive norm assignment at design time.

In Tamargo et al. [12], knowledge dynamics in agents' belief based on a collaborative multi-agent system was examined. Four change operators were introduced: expansion, contraction, prioritized revision, and non-prioritized revision. For all of them, both constructive definitions and an axiomatic characterization by representation theorems were given. Minimal change, consistency maintenance, and non-prioritization principles were formally justified by the researchers. Khan and Lespérance [13] and Riemsdijk et al. [14] also contributed in the area of agents' beliefs and goals changing. Khan and Lespérance [13] in their work ensured that the agent's chosen goals/intentions were consistent with each other and with the agent's knowledge. When the environments change, the agents recomputed their chosen goals and some inactive goals may become active again. This ensured that the agent maximized utility. Riemsdijk et al. [14] gave a formal and generic operationalization of goals by defining an abstract goal architecture, which described the adoption, pursuit, and dropping of goals in a generic way.

In our work, we considered that adaptation could be provided at both the agent level and the algorithm level; but, in this paper, emphasis was placed on adaptivity of the mining agents.

III. SYSTEM DESIGN

This design of the proposed adaptive mining agent architecture is presented in this section.

A. The Proposed Adaptive Architecture

The adaptive architecture described in this work was based on the earlier Distributed Association Rule Mining (DARM) architecture AMAARMD presented by Ogunde et al. [15]. The architecture was characterized by a given distributed data mining task being executed in its entirety using the mobile agents. In general, this was expressed as m mobile agents traversing n data sources (where m >n).

B. The Adaptive Algorithm

In our architecture, the very first mining by the system is based on the traditional Apriori algorithm (if the initial

dataset is very small) and the Partition Enhanced Algorithm (PEA), which is an improved version of the state-of-the-art Apriori algorithm contributed by the researchers. PEA partitions the large dataset into smaller partitions, while mining each partition (as it easily fits into the memory) to generate local patterns, which was integrated to generate global frequent patterns for a particular large data site in the DARM architecture. The partition sizes were chosen such that each partition can be accommodated in the main memory, so that the partitions are read only once in each phase. In this work, the mining agent examines the system to obtain the current total available memory space and then use this information to divide DB into the several partitions. This is to ensure that each partition fits into the main memory during the first phase of the mining. Subsequent mining of the incremental database is done with the Adaptive Incremental Mining (AIM) algorithm also contributed by the researchers. Details of *PEA* and *AIM* algorithms were not presented as this particular work is focused on the adaptivity of DARM agents. *AIM* mines only the incremental database dynamically whenever there is a pre-defined increase in the total transactions inside the database. It stores the previously frequent and non-frequent itemsets to be able to determine whether an itemset is still frequent in the updated database or it is no more frequent, taking note of the specific time and periods when these changes occurred for proper management decisions by data miners.

C. Description of the Adaptive DARM Agents

In this section, the different types of agents and users in the architecture are described. Agent types: User Agent (UA), Association Rule Mining Coordinating Agent (ARMCA), Data Source Agent (DSA), Mobile Agent Based Association Rule Miner (MAARM), Mobile Agent- Based Result Reporter (MARR), Results Integration Coordinating Agent (RICA), Task Agent (TA) and Registration Agent (RA). All agents are created and resident in the DARM server. The UA and DSA are interface agents because they all provide “interfaces” to either users, or data sources. UA provides the interface between the architecture, users and the rest of the architecture; while DSA provides the interface between input data and the rest of the architecture. MAARM agents are processing agents because they carry out the required ARM at the data sites automatically or in response to user requests, and possibly, to pre-process data within the system. A description of the various agents in the system described and their interactions are summarized in Figure 1.

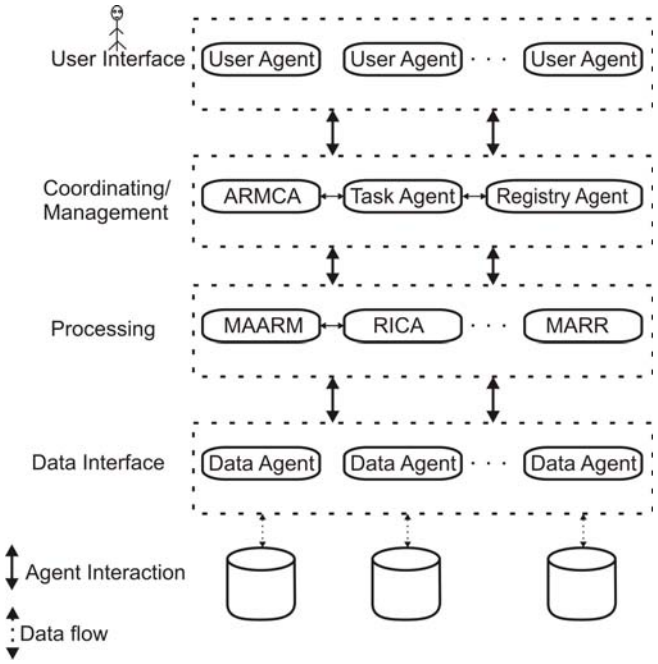


Figure 1: Agent Architecture for the System.

According to Figure 1, the task agent receives a DARM request and asks ARMCA to check all available databases (DA) and MAARM agents to find: (i) which data to use, and (ii) which data mining algorithms held by ARM agents are appropriate. The RA informs all appropriate agents that are already in the system and interested of a new agent arrival. When any new agent is introduced into the system, the TA then passes the DARM task to MAARM agents, which clones itself into multiple copies depending on the number of available data sites, and then travel to each data site in the DARM task. Each data site must have an interface agent – data agent (DA) to check the database for a matching schema and then report back to the MAARM agent. Distributed association rule mining at each data source is performed by the ARM agents – MAARMs. The return of results information at each data site is carried out by the MARRs. The agent RICA integrates the various local results to get a global rule from the DARM sites. Final results or knowledge are passed from the RICA to the TA and then to the UA all through the DARM server.

D. Adaptive Mobile Agent Association Rule Miner (AMAARM)

An agent can be viewed as software satisfying an ordered set of goals to achieve some overall objective. The agent takes a sequence of action in order to satisfy the next goal in the set. Adaptation can be viewed as changing the goal set. The effect of the change can be a new set of actions to achieve the same overall objective as before, or it may even result in a new overall objective if the original objective cannot be achieved anymore in the current environment. The model described here consists of two components: a

 ISBN: 978-1-61208-219-6

Mechanism and an Adapter. The Mechanism is the interface of the AMAARM to the environment. It contains sensors that periodically sense the DARM environment parameters and report their findings to the Mechanism.

For instance, if a particular data server is down out of five data sites or there is a sort of interference to the mining process, the mechanism of the mining agent senses this and reports it to the Adapter, which decides a waiting period for the agent in order to make another attempt to complete the mining process or in the worst case excludes the result of that particular site from the global knowledge integration performed by RICA after a number of preset trial-times. This means that if the set goal for RICA was to integrate local ARM results from five data sites, it will now change the set goal to integrate only the four available results, which are returned as the global knowledge.

It also contains actors that can take actions to change the environment the mining agent is in. The Adapter is the component that decides whether adaptation is necessary or not. If adaptation is necessary, the adapter determines how best to adapt to the current environment. The Mechanism senses the environment through the sensors, analyze them, and create a view of the environment called a percept. The percept is passed on to the adapter, which then decides whether adaptation is necessary or not. Another instance of an unforeseen problem that can arise here are the possibilities of collision of the mining agent with either other mining agents or agents carrying out some other tasks within the same environment. As a matter of fact, all these agents could be possibly competing for the same resources and these could hamper the performance of the association rule mining agent in such environments, hence there is a need for adaption on the fly by these agents. Therefore, if adaptation is needed, a new set of goals is passed on to the mechanism, which then transforms the set of goals into a set of actions to be carried out, and then carries out the actions. The actors are used to make any environment change specified in an action. Figure 2 shows the basic structure of the AMAARM components and their interactions as explained above.

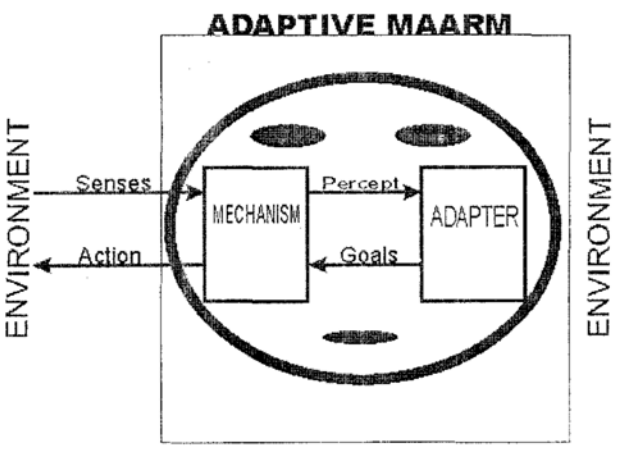


Figure 2: Components of AMAARM

In figure 2, the mechanism actually senses the DARM environment, creates a percept for the adapter, which decides the action, that is, whether initial the goals of the mining agent should be maintained or changed.

E. Description of the AMAARM's Mechanism

The state of the AMAARM's Mechanism is represented by the 3-tuple <S, L, T>, where S represent the behavioral state of the Mechanism which identifies what the Mechanism is doing currently. *S* could therefore be the state where the Mechanism senses the environment for changes or the state at which action is taken to change the agent's goal. *L* is the current location of the agent, and *T* is the time the Mechanism had spent in its current state. The state variable *S* can take one of three possible values: extractGoal, executeCommand and senseEnvironment. In the extractGoal behavioral state, the Mechanism picks up the current mining goals to be executed, and generates the set of commands for it. In the executeCommand behavioral state, the generated commands are carried out. In the senseEnvironment behavioral state, the Mechanism senses the environment and forms a current view of the environment, and then passes it to the Adapter. Figure 3 is a state diagram showing the different states that the AMAARM's Mechanism can be at any point in time and the possible state transitions.

According to Figure 3, the Mechanism can always be in any of these three states: the FirstNormalState = <extractGoal, L, T> or SecondNormalState = <executeCommand, L, T> or SenseEnvironmentState = <senseEnvironment, L, T>, where *L* and *T* retains their predefined definitions as *L* contains the current location of the agent, and *T* is the time the MAARM had been in that state, which is usually reset to 0 every time a state transition occurs. Initially, the mechanism enters the FirstNormalState on receiving an ordered set of goals from the adapter. The commands for the next goal in the ordered list are generated and a transition to state SecondNormalState occurs.

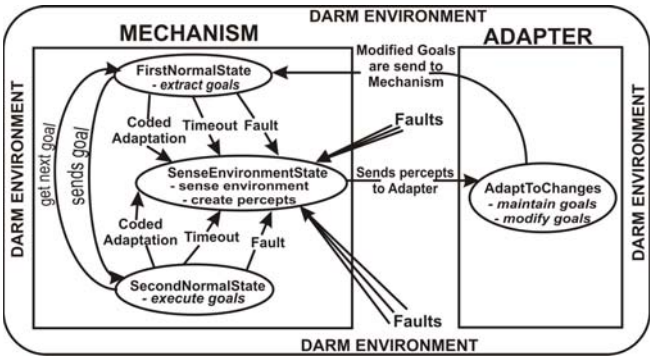


Figure 3: State Diagram of AMAARM

In the SecondNormalState, the commands are executed, and then, the transition goes back to the FirstNormalState in order to generate the commands for the next goal in the list of goals. This process continues until all the goals in the list are executed by the Mechanism. However, the Mechanism may go to the third state, that is, SenseEnvironmentState = <senseEnvironment, L, T> from either the FirstNormalState

or the SecondNormalState if any of the following happens: a timeout, a fault, or a coded adaptation (an explicit command in the application code itself to sense the environment for adaptation reasons), a collision, an attempt to corrupt the mining agent, etc. The *T* component of the Mechanism state detects a timeout if an action is not carried out within a specified time in the SecondNormalState. This may indicate changed environment and may force AMAARM to sense the environment and determine whether adaptation is necessary.

In the SenseEnvironmentState, the environment is usually sensed for any sort of change earlier mentioned and a percept is sent to the Adapter to see if any adaptation is necessary. Normally, the environment can also be sensed in the FirstNormalState and the SecondNormalState, occasionally as the case may be; but, in these cases there is usually no interaction with the Adapter, therefore the values sensed are usually used internally by those states of the AMAARM.

Given a DARM environment, there may be different ways the mining agent can adapt. Thus some type of ranking of the adaptation methods in the adaptation policy is necessary. This is achieved by a motivation degree function. Motivation is any desire or preference that that can lead to the generation and adoption of goals and which affects the outcome of the reasoning or behavioral task intended to satisfy the goal [16]. A motivation degree is therefore associated with each adaptation method, which is the probability of success in achieving the final goal if the set of goals corresponding to the adaptation method is selected as the current set of goals. The Adapter then selects the adaptation method with the highest motivation degree corresponding to the current environment. The set of adaptation methods and the motivation degree function can be hard-coded or learnt dynamically from history or a combination of both where the user specifies an adaptation policy and a motivation degree function, which then can be modified dynamically as well. For the purpose of this work, the adaptation policies for AMAARM are hard-coded.

The adaptive state of the AMAARM is thus described by the 3-tuple <MS, AS, AS>, where MS is the Mechanism state, AS is the Adapting state, and AS is the Application-specific state for the AMAARM. On receiving a percept from the Mechanism, the Adapter goes through the set of adaptation methods, looking for the ones that match the percept. The one with the highest motivation degree is then chosen, and the current set of ARM goals are modified to be the one corresponding to that adaptation method. The new ARM goal set is passed to the Mechanism, which then generates and executes commands for the set of goals. If no adaptation method matches the current environment, adaptation is deemed unnecessary and no change to the goal set occurs. Thus, no adaptation can also be viewed as a special adaptation method.

F. Description of the AMAARM's Adapter

The Adapter state consists of the two tuple <S, T>, where S is the behavioral state of the Adapter, which can only adapt by either maintaining the mining goals, if a change of goal is not necessitated by the percepts received from the mechanism, or modify the mining goals if the percepts received from the mechanism is significant for modifying the original goals. *T* is the time spent in the *AdaptToChange* state. An *attribute* is a perceivable feature of the DARM environment, e.g., a fault in the DARM environment, a time-out, agent collision or an attack or violation of the integrity of the mining agents. A *percept* is a set of attributes, that is, a view of the DARM environment. An *adaptation method* is a single mapping from a percept to a set of mining goals. An *adaptation policy* is a set of adaptation methods. Thus, in this case, the adaptation policy specifies the possible ways in which the AMAARM adapts to different DARM environments.

IV. CONCLUSION AND FUTURE WORK

An adaptive distributed association rule mining architecture with adaptive mining agents was presented. The system described here promises to guarantee the completion of major DARM tasks even in the face of unforeseen circumstances and faults. Each individual data server has some specific data and resource requirements, all of which have to be satisfied before the task can be started. An adaptive mining agent AMAARM executing a task migrates to a data server from the DARM server, and tries to generate the frequent itemsets. If all the necessary resources at the data site are available and the environment is conducive, then the mining task is executed. Otherwise, the data server environment is sensed to get an idea about the time the adaptive mining agent may have to wait to perform the mining task. The MAARM relies on the coded adaptation to make this adaptation decision. Future work will consider the set of adaptation methods and policies in DARM that will be a combination of hard-coded policies and also dynamic learning of earlier mining agents' adaptation from history. Implementation of the system using synthetic and real life datasets in order to test the performance of this method will also be done as a future work.

REFERENCES

[1] R. Agrawal, T. lmielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," In Proceedings of the ACM SIGMOD Conference on Management of Data, Washington, D.C, 1993, pp. 207-216.

[2] V. S. Rao and S. Vidyavathi, "Distributed Data Mining And Mining Multi-Agent Data," (IJCSE) International Journal on Computer Science and Engineering vol. 02, no. 04, 2010, pp. 1237-1244.

[3] M. Z. Ashrafi., D. Taniar, and K. Smith, Monash University "ODAM. An Optimized Distributed Association Rule Mining Algorithm", IEEE distributed systems online, pp. 1541-4922 © 2004 published by the IEEE computer society vol. 5, no. 3; march 2004

[4] H. Kargupta, H. Ilker, and S. Brian, "Scalable, istributed data mining-an agent architecture," In Heckerman et al. [8], pp. 211.

[5] A. O. Ogunde, O. Folorunso, A. S. Sodiya, and G. O. Ogunleye, "A Review of Some Issues and Challenges in Current Agent-Based Distributed Association Rule Mining," Asian Journal of Information Technology, vol. 10, no. 02, 2011, pp. 84-95.

[6] K. A. Albashiri, "EMADS: An Investigation into the Issues of Multi-Agent Data Mining," PhD Thesis, The University of Liverpool, Ashton Building, United Kingdom, 2010. www.csc.liv.ac.uk/research/techreports/tr2010/ulcs-10-004.pdf [retrieved: June, 2012]

[7] M. Wooldridge, “An Introduction to Multi-Agent Systems,” John Wiley and Sons (Chichester. United Kingdom), 2009.

[8] M. Wooldridge, “Reasoning About Rational Agents,” Cambridge, MA: MIT Press, 2000.

[9] F. M. T. Brazier., and N. J. E. Wijngaards, “Automated servicing of agents,” AISB Journal, vol.1, no.1, pp. 5-20, 2001.

[10] S. Ranjan, A. Gupta, A. Basu, A. Meka, and A.Chaturvedi, “Adaptive mobile agents: Modeling and a case study”. 2nd Workshop on Distributed Computing “IEEE md CFP : WDC’2000’.

[11] N. Lacey and H.Hexmoor, “Norm Adaptation and Revision in a Multi-Agent System”, American Association of Artificial Intelligence, FLAIRS 2003, pp. 27-31.

[12] L. H. Tamargo, A. J. Garcia, M. A. Falappa, and G. R. Simari, “Modeling knowledge dynamics in multi-agent systems based on informants,” The Knowledge Engineering Review, Cambridge University Press, DOI: 10.1017/S000000000000000, Printed in the United Kingdom, vol. 00:0, pp. 1-31, 2010.

[13] S. M. Khan and Y. Lespérance, “A Logical Framework for Prioritized Goal Change,” Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), May, 10–14, 2010, Toronto, Canada, pp. 283-290.

[14] M. B. Riemsdijk, M. Dastani, and M. Winikoff, “Goals in Agent Systems: A Unifying Framework,” Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), May, 12-16, 2008, Estoril, Portugal, pp. 713-720.

[15] A. O. Ogunde, O. Folorunso, A. S. Sodiya, and J. A. Oguntuase, “Towards an adaptive multi-agent architecture for association rule mining in distributed databases,” Adaptive Science and Technology (ICAST), 2011 3rd IEEE International Conference on 24-26 Nov. 2011, pp. 31 – 36 from IEEE Xplore.

[16] Z. Kunda, “The case for motivated reasoning,” Psychological Bulletin, 108 (3), pp. 480-498, 1990.

 ISBN: 978-1-61208-219-6

Adaptive Control of a Biomethanation Process using Neural Networks

Dorin Sendrescu
Department of Automatic Control
University of Craiova
Craiova, Romania
e-mail: dorins@automation.ucv.ro

Elena Bunciu
Department of Automatic Control
University of Craiova
Craiova, Romania
e-mail: selena@automation.ucv.ro

***Abstract*— This paper deals with the design of an adaptive control scheme for the regulation of the acetate concentration in a biomethanation process with production of methane gas that takes place inside a Continuous Stirred Tank Bioreactor. The control structure is based on the nonlinear model of the process whose parameters are identified using the distributions method and the unknown reaction rates are estimated using a radial basis neural network. These estimations are then used in a nonlinear model predictive control (NMPC) scheme. Minimization of the cost function is realized using the Levenberg–Marquardt numerical optimization method. The effectiveness and performance of the proposed control strategy is illustrated by numerical simulations. The simulation results obtained with a continuous stirred tank reactor plant model confirmed the good quality of the control.**

Keywords-neural networks; biotechnological process; adaptive control

I. INTRODUCTION

In the last period, the control of biotechnological processes has been an important problem attracting wide attention. The main engineering motivation in applying advanced control methods to such processes is to improve operational stability and production efficiency. But, the use of modern control for these bioprocesses is still low. The nonlinearity of the bioprocesses and the uncertainty of kinetics impose the adaptive control strategy as a suitable approach. So, the difficulties encountered in the measurement of the state variables of the bioprocesses impose the use of the so-called "software sensors". Note that these software sensors are used not only for the estimation of the concentrations of some components but also for the estimation of the kinetic parameters or even kinetic reactions. The adaptive control scheme used in this work is based on a predictive controller that uses the nonlinear dynamic model to predict the effect of sequences of control steps on the controlled variables.

This paper deals with the adaptive control of a wastewater biodegradation process. The dynamics of this biotechnological process are described by a set of nonlinear differential equations obtained from the reaction scheme and the unknown reaction rates are estimated using a radial basis neural network. For the estimation of unknown parameters of the process, the distribution approach is used. The parameter identification of deterministic nonlinear continuous-time systems (NCTS), modeled by polynomial type differential equation has been considered by numerous authors [1], [2]. In this paper, we use an identification method for a class of NCTS considering that the unknown parameters can appear in rational relations with measured variables. Using techniques used in distribution approach, the measurable functions and their derivatives are represented by functionals on a fundamental space of testing functions. Such systems are common in biotechnology [3]. The main idea is to use a hierarchical structure of identification. First, some state equations are utilized to obtain a set of linear equations in some parameters. The results of this first stage of identification are utilized for expressing other parameters by linear equations. This process is repeated until all parameters are identified.

The paper is organized as follows. Section II is devoted to description and modeling of a biomethanation process. The adaptive control strategy is presented in Section III. Simulations results presented in Section IV illustrate the performance of the proposed control algorithms and, finally, Section V concludes the paper.

II. PROCESS MODELING

A. Analytical approach of process modeling

Anaerobic digestion is a multi-stage biological wastewater treatment process whereby bacteria, in the absence of oxygen, decompose organic matter to carbon dioxide, methane and water. A linear model, no matter how well it has been structured and tuned, may be acceptable only in the case where the system is working around the operating point. If the system is highly nonlinear, such as biotechnological processes, control based on the prediction from a linear model may result in unacceptable response. In some cases, remarkable static errors exist, and in other cases, oscillation or even instability may occur. Therefore, some kinds of non-linear models should be used to describe the behavior of a highly non-linear system [4].

In this paper, we consider a biomethanation process – wastewater biodegradation with production of methane gas that takes place inside a Continuous Stirred Tank Bioreactor whose reduced model is presented in [5]. In the first phase, the glucose from the wastewater is decomposed in fat volatile acids (acetates, propionic acid), hydrogen and

 ISBN: 978-1-61208-219-6

inorganic carbon under action of the acidogenic bacteria. In the second phase, the ionised hydrogen decomposes the propionic acid CH_3CH_2COOH in acetates, H_2 and carbon dioxide CO_2. In the first methanogenic phase, the acetate is transformed into methane and CO_2, and finally, in the second methanogenic phase, the methane gas CH_4 is obtained from H_2 and CO_2, [5]. The following simplified reaction scheme is considered,

$$\begin{cases} S_1 \xrightarrow{\Phi 1} X_1 + S_2 \\ S_2 \xrightarrow{\Phi 2} X_2 + P_1 \end{cases} \tag{1}$$

where: S_1 represents the glucose substrate, S_2 the acetate substrate, X_1 is the acidogenic bacteria, X_2 the acetoclastic methanogenic bacteria and P_1 represents the product, i.e. the methane gas. The reaction rates are denoted by Φ_1, Φ_2. The corresponding dynamical model is:

$$\frac{d}{dt}\begin{bmatrix} X_1 \\ S_1 \\ X_2 \\ S_2 \\ P_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -k_1 & 0 \\ 0 & 1 \\ k_2 & -k_3 \\ 0 & k_4 \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} - D \begin{bmatrix} X_1 \\ S_1 \\ X_2 \\ S_2 \\ P_1 \end{bmatrix} + \begin{bmatrix} 0 \\ DS_{in} \\ 0 \\ 0 \\ -Q_1 \end{bmatrix} \tag{2}$$

where S_{in} is the influent substrate, Q_1 the methane gas outflow rate, D the dilution rate and the state vector of the model is:

$$\xi = [X_1\, S_1\, X_2\, S_2\, P_1]^T = [\xi_1\, \xi_2\, \xi_3\, \xi_4\, \xi_5]^T \tag{3}$$

whose components are concentrations in (*g/l*). The reaction rates are nonlinear functions of the state components, expressed as

$$\Phi = \Phi(\xi) = \begin{bmatrix} \Phi_1(\xi) \\ \Phi_2(\xi) \end{bmatrix} \tag{4}$$

The reaction rates for this process are given by the Monod law [3]

$$\Phi_1(\xi) = \mu_1 \frac{S_1 \cdot X_1}{K_{M_1} + S_1} \tag{5}$$

and the Haldane kinetic model [3]:

$$\Phi_2(\xi) = \mu_2 \frac{S_2 \cdot X_2}{K_{M_2} + S_2 + S_2^2 / K_i} \tag{6}$$

where K_{M_1}, K_{M_2} are Michaelis-Menten constants, μ_1, μ_2 represent specific growth rates coefficients and K_i is the inhibition constant.

For simplicity, shall we denote the plant parameters by the vector:

$$\theta = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7 \quad \theta_8 \quad \theta_9]^T \tag{7}$$

where:

$$\theta_1 = k_1; \quad \theta_2 = k_2; \quad \theta_3 = k_3; \quad \theta_4 = k_4; \quad \theta_5 = \mu_1; \\ \theta_6 = \mu_2; \theta_7 = K_{M_1}; \theta_8 = K_{M_2}; \theta_9 = K_i \tag{8}$$

Because the dilution rate D can be externally modified, it will be considered the third component of the input vector

$$u = [u_1 \quad u_2 \quad u_3]^T \tag{9}$$

The other two components of u are the concentration S_{in} and the methane gas outflow rate Q_1 so,

$$u_1 = S_{in}; u_2 = Q_1; u_3 = D \tag{10}$$

Usually, Q_1 depends on state variables, $Q_1 = \psi(\xi)$; determining a feedback to the input u_2. Written explicitly by components, the state equation (2), within the above notations, takes the form:

$$\dot{\xi}_1 = \Phi_1 - u_3 \cdot \xi_1 \tag{11}$$

$$\Phi_1 = \theta_5 \cdot \frac{\xi_1 \cdot \xi_2}{\theta_7 + \xi_2} \tag{12}$$

$$\dot{\xi}_2 = -\theta_1 \cdot \Phi_1 - u_3 \cdot \xi_2 + u_1 \cdot u_3 \tag{13}$$

$$\dot{\xi}_3 = \Phi_2 - u_3 \cdot \xi_3 \tag{14}$$

$$\Phi_2 = \theta_6 \cdot \frac{\xi_3 \cdot \xi_4}{\theta_8 + \xi_4 + \theta_9^{'} \cdot \xi_4^2}, \theta_9^{'} = \frac{1}{\theta_9} \tag{15}$$

$$\dot{\xi}_4 = \theta_2 \cdot \Phi_1 - \theta_3 \cdot \Phi_2 - u_3 \cdot \xi_4 \tag{16}$$

$$\dot{\xi}_5 = -u_3 \cdot \xi_5 + \theta_4 \cdot \Phi_2 - u_2 \tag{17}$$

B. Parameters estimation

For parameters estimation the distribution based method was used. In this approach the set of nonlinear differential equations describing the state evolution is mapped into a set of linear algebraic equations respect to the model parameters. Using techniques utilized in distribution approach, the measurable functions and their derivatives are represented by functionals on a fundamental space of testing functions. The main advantages of this method are that a set of algebraic equation with real coefficients results and the formulations are free from boundary conditions.

Let Φ_n be the fundamental space from the distribution theory of the real functions $\varphi : R \to R, t \to \varphi(t)$ and $q : R \to R, t \to q(t)$ a function which admits a Riemann integral on any compact interval T from **R**. Using this function, a unique distribution

$$F_q : \Phi_n \to R, \varphi \to F_q(\varphi) \in R \tag{18}$$

can be built by the relation:

$$F_q(\varphi) = \int_R q(t)\varphi(t)dt, \forall \varphi \in \Phi_n \tag{19}$$

In distribution theory, the notion of k-order derivative is introduced. If $F_q \in \Phi_n$, then its k-order derivative is a new distribution $F_q^{(k)} \in \Phi_n$ uniquely defined by the relations:

$$F_q^{(k)}(\varphi) = (-1)^k F_q(\varphi^{(k)}),\ \forall \varphi \in \Phi_n \tag{20}$$

$$\varphi \to F_q^{(k)}(\varphi) = (-1)^k \int_R q(t)\varphi^{(k)}(t)dt \in R \tag{21}$$

where

$$\varphi^{(k)} : R \to R,\ t \to \varphi^{(k)}(t) = \frac{d^k \varphi(t)}{dt^k} \tag{22}$$

is the *k*-order time derivative of the testing function.

When $q \in C^k(R)$, then

$$F_q^{(k)}(\varphi) = \int_R q^{(k)}(t)\varphi(t)dt = (-1)^k \int_R q(t)\varphi^{(k)}(t)dt, \tag{23}$$

that means the k-order derivative of a distribution generated by a function $q \in C^k(R)$ equals to the distribution generated by the k-order time derivative of the function *q*.

So, in place of the states and their time derivatives of a system one utilizes the corresponding distributions and, in some particular cases, it is possible to obtain a system of equations linear in parameters. If the system is compatible then all the model parameters are structurally identifiable.

Consider all state variables accessible for measurements. The dynamical system (11)-(17) contains rational dependencies between parameters and measured variables. To obtain linear equations in unknown parameters, the identification problem is split in several simpler problems. Based on the specific structure of this system, it is possible to group the state equations, in such way to determine five interconnected relations. They are organized in a hierarchical structure. In the first stage, some state equations are utilized to obtain a set of linear equations in some parameters. If these parameters are identified then they can be used as known parameters in the following stages. This process is repeated in the other stages until all the parameters are identified.

III. ADAPTIVE CONTROL ALGORITHM

A. Problem formulation

Consider the following discrete-time, time-invariant nonlinear system:

$$\xi_{k+1} = f(\xi_k, u_k) \tag{24}$$

with ξ_k the state vector, u_k the control signal, corresponding in our case with the discretisation of system (2). The objective is to regulate the state vector (or a particular output signal) to a specified setpoint value while guaranteeing that certain input and state constraints:

$$\begin{aligned} \xi_{\min} \le \xi_k \le \xi_{\max} \\ u_{\min} \le u_k \le u_{\max} \end{aligned} \tag{25}$$

Nonlinear model predictive control treats such a constrained control problem by repeatedly solving the following optimization problem [6]:

$$\min \sum_{i=0}^{N-1} (\xi_{ref} - \xi_{k+i})^T Q(\xi_{ref} - \xi_{k+i}) + u_{k+i}^T R u_{k+i} \tag{26}$$

$$s.t. \begin{cases} \xi_{k+1} = f(\xi_k, u_k) \\ \xi_{\min} \le \xi_k \le \xi_{\max} \\ u_{\min} \le u_k \le u_{\max} \end{cases} \tag{27}$$

Among the whole sequence resulting from the on-line optimization, only the first optimal control is applied as input to the system [7]. At the next sampling time, the current state is obtained (measured or estimated) and the optimization problem (26), (27) is solved again with this new initial state value, according to the well-known receding horizon principle.

B. The Neural Network Model

The predictive model for a conventional MPC controller is usually a linear model which is preferred as being more intuitive and requiring less a prior information for its identification. MPC based on linear models is acceptable if the process operates at a single setpoint and the primary use of the controller is the rejection of disturbances. Many chemical processes, including polymer reactors, do not operate at a single setpoint. However, these models are not suitable for a nonlinear system such as biotechnological processes. To solve this problem neural networks are proposed to obtain the estimated output used by the MPC controller, because the neural networks have the ability to map any nonlinear relationships between an input and output set. There have also been many reports on the application of neural network to bioprocesses control, modelling and identification [8], [9].

In this paper, the process model is obtained using a radial basis neural network (RBNN) with adjustable parameters to approximate the reaction rates Φ_1 and Φ_2 from model (2). A RBNN is made up of a collection of $p > 0$ parallel processing units called nodes. The output of the *i*th node is defined by a Gaussian function $\gamma_i(x) = \exp(-|x - c_i|^2 / \sigma_i^2)$, where $x \in \Re^n$ is the input to the NN, c_i is the centre of the *i*-th node, and σ_i is its size of influence. The output of a RBNN, $y_{NN} = F(x,W)$, may be calculated as [13]

$$F(x,W) = \sum_{i=1}^{p} w_i \gamma_i(x) = W^T(t)\Gamma(x) \tag{28}$$

where $W(t) = [w_1(t)\ w_2(t) \dots w_p(t)]^T$ is the vector of network weights and $\Gamma(x)$ is a set of radial basis functions defined by $\Gamma(x) = [\gamma_1(x)\ \gamma_2(x) \dots \gamma_p(x)]^T$.

Given a RBNN, it is possible to approximate a wide variety of functions $f(x)$ by making different choices for *W*. In particular, if there is a sufficient number of nodes within

 ISBN: 978-1-61208-219-6

the NN, then there is some W^* such as $\sup_{x \in S_x} \left| F(x, W^*) - f(x) \right| < \varepsilon$, where S_x is a compact set, $\varepsilon > 0$ is a finite constant, provided that $f(x)$ is continuous. The RBNN is used to estimate the reaction rates Φ_1 and Φ_2 (that are considered unknown) using some state measurements.

C. The Neural Network Adaptive Control Algorithm

The model predictive control is a strategy based on the explicit use of system model to predict the controlled variables over a certain time horizon, called the prediction horizon. The adaptive control strategy can be described as follows:

1) Using the on-line measurements the unknown dynamics of the system are estimated using an ANN.

2) At each sampling time, the value of the controlled variable y_{t+k} is predicted over the prediction horizon $k=1,..., N_y$. This prediction depends on the future values of the control variable u_{t+k} within a control horizon $k=l,..., N_u$.

3) A reference trajectory y^{ref}_{t+k}, $k=1,.., N$ is defined which describes the desired system trajectory over the prediction horizon.

4) The vector of future controls u_{t+k} is computed such that an objective function (a function of the errors between the reference trajectory and the predicted output of the model) is minimised.

5) Once the minimisation is achieved, the first optimised control actions are applied to the plant and the plant outputs are measured. These measurements are used as the initial states of the model to perform the next iteration. Steps 1 to 5 are repeated at each sampling instant.

The adaptive control strategy is illustrated by the scheme represented in Fig. 1.

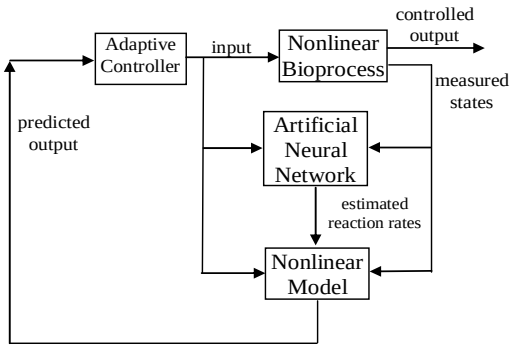


Figure 1. Adaptive control scheme.

When a solution of the nonlinear least squares (NLS) minimization problem cannot be obtained analytically, the NLS estimates must be computed using numerical methods. To optimize a nonlinear function, an iterative algorithm starts from some initial value of the argument in that function and then repeatedly calculates next available value according to a particular rule until an optimum is reached approximately. Between many different methods of numerical optimization the Levenberg-Marquardt (LM) algorithm was chosen. The LM algorithm is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real-valued functions [10]. It has become a standard technique for non-linear least-squares problems, widely adopted in a broad spectrum of disciplines. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method. When the current solution is close to the correct solution, it becomes a Gauss-Newton method.

IV. SIMULATION RESULTS

In this section, we will apply the designed adaptive control in the case of the anaerobic digestion bioprocess presented in Section II. In order to control the output pollution level y, as input control we chose the dilution rate, $u = D$. The main control objective is to maintain the output y at a specified low level pollution $y_d \in \Re$. We will analyze the realistic case where the structure of the system of differential equation (2) is known and specific reaction rates Φ_1 and Φ_2 (described by "(6)" and "(7)") are completely unknown and must be estimated. Using a RBNN from subsection 3.2, one constructs an on-line estimate of Φ_1 respectively of Φ_2.

The performance of the adaptive controller presented in Subsection III - C has been tested through extensive simulations by using the process model (2). The values of yield and kinetic coefficients are: k_1 = 3.2, k_2 = 16.7, k_3 = 1.035, k_4 = 1.1935, k_5 = 1.5, k_6 = 3, k_7 = 0.113, μ_1^* = 0.2 h^{-1}, K_{M_1} = 0.5 g/l, μ_2^* = 0.35 h^{-1}, K_{M_2} = 4 g/l, K_{I_2} = 21 g/l, and the values α_1 = 1.2, α_2 = 0.75. It must be noted that for the reaction rates estimation a full RBNN with deviation $\sigma_i = 0.05$ was used. The centres c_i of the radial basis functions are placed in the nodes of a mesh obtained by discretization of states $X_1 \in [1, 12]$ g/l, $X_2 \in [0.4, 0.65]$ g/l, $S_1 \in [0.1, 1.4]$ g/l and $S_2 \in [0.3, 1.6]$ g/l with $dX_i = dS_i = 0.2$ g/l, $i = 1, 2$.

The simulation results, obtained with a sample period T_s=6 min, are presented in Figs. 2 – 5. In Fig. 2 the controlled output trajectory is presented and in Fig. 3 the nonlinear model predictive control action (dilution rate D evolution) is depicted. The functions Φ_1 and Φ_2 provided by the RBNN are depicted versus the "real" functions in Fig. 4 and Fig. 5. From these figures it can be seen that the behaviour of the control system with adaptive controller is very good, although the process dynamics are incompletely known. The control action has an oscillatory behavior, but these oscillations are relatively slow and with small magnitude.

 ISBN: 978-1-61208-219-6

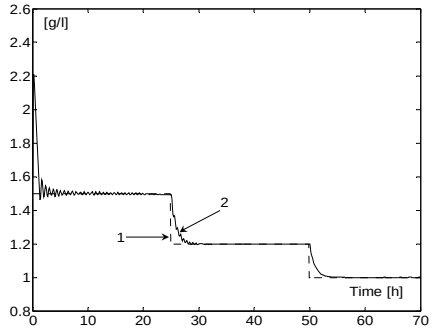


Figure 2. The controlled output evolution (reference (1) and controlled output (2)).

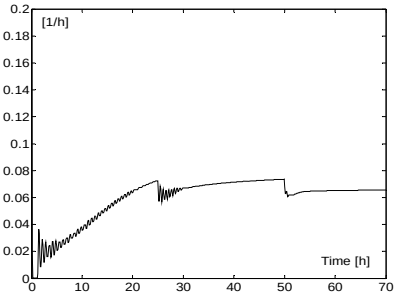


Figure 3. The nonlinear model predictive control action (dilution rate D).

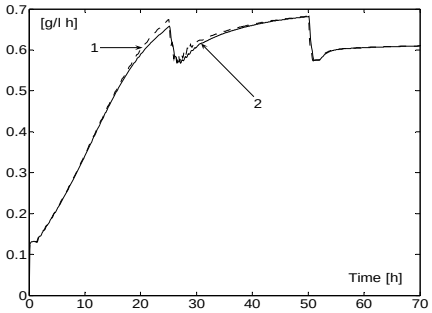


Figure 4. The real reaction rate Φ_1 (1) versus the function provided by the RBNN (2)

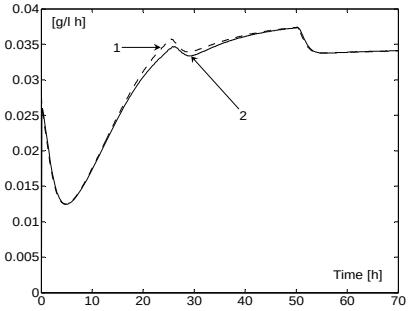


Figure 5. The real reaction rate Φ_2 (1) versus the function provided by the RBNN (2)

V. CONCLUSIONS

In this paper, an adaptive control strategy was developed for a wastewater treatment bioprocess. The nonlinear model used by the control algorithm was obtained using the analytical description of the biochemical reactions. The yield coefficients are identified using an algorithm based on test functions from distribution theory. This procedure is a functional type method, which transforms a differential system of equations to an algebraic system in unknown parameters. The relation between the state variables of the system is represented by functionals using techniques from distribution theory based on test functions from a finite dimensional fundamental space. The identification algorithm has a hierarchical structure, which allows obtaining a linear algebraic system of equations in the unknown parameters. The unknown reaction rates are estimated using radial basis neural networks. The nonlinear model states are used to calculate the optimal control signal applied to the system. The optimization problem was solved using the iterative Levenberg-Marquardt algorithm. The efficiency of the proposed algorithm was illustrated by numerical simulation.

ACKNOWLEDGMENT

This work was supported by the National University Research Council - CNCSIS, Romania, under the research project TE 106, no. 9/04.08.2010.

REFERENCES

[1] A. Patra and H. Unbehauen, “Identification of a class of nonlinear continuous time systems using Hartley modulating functions”, International Journal of Control, vol. 62, no. 6, pp. 1431-1452, 1995.

[2] A. Pearson and F. Lee, “On the identification of Polynomial input-output differential systems”, IEEE Transactions on Automatic Control, vol. AC30, no. 8, pp. 778-782, 1985.

[3] G. Bastin and D. Dochain, On-line Estimation and Adaptive Control of Bioreactors, Elsevier, Amsterdam, 1990.

[4] D. Dochain and P. Vanrolleghem, Dynamical Modelling and Estimation in Wastewater Treatment Processes, IWA Publishing, 2001.

[5] E. Petre and D. Selisteanu, Modelling and Identification of Depollution Bioprocesses, Universitaria, Craiova, 2005.

[6] M. Cannon, B. Kouvaritakis, Y.I. Lee and A.C. Brooms,” Efficient non-linear model based predictive control”, International Journal of Control, vol. 74, no. 4, pp. 361-372, 2001.

[7] E.F Camacho and C. Bordons, Model Predictive Control, second edition, Springer, 2004.

[8] J.W. Eaton and J.R. Rawlings, “Feedback Control of Nonlinear Processes Using Online Optimization Techniques,” Computers and Chemical Engineering, vol. 14, pp. 469-479, 1990.

[9] J.T. Spooner and K.M. Passino, “Decentralized adaptive control of nonlinear systems using radial basis neural networks”, IEEE Transactions on Automatic Control, vol. 44, no. 11, pp. 2050-2057, 1999.

[10] J. Nocedal and S.J. Wright, Numerical Optimization, Springer, New York, 1999.

 ISBN: 978-1-61208-219-6

Adaptive Control for a De-pollution Bioprocess

Bunciu (Stanciu) Elena, Şendrescu Dorin, Petre Emil
Department of Automatics, Electronics and Mechatronics
University of Craiova
Craiova, Romania
bunciu.elena@yahoo.com, dorins@automation.ucv.ro, epetre@automation.ucv.ro

Abstract— **This paper deals with the design of an interval observer for estimation of some state variables in an uncertain biotechnological process. The observer is applied to estimate the biomass concentration in an anaerobic bioprocess with a simplified mathematical model. In this approach only the bounds of the biomass concentration and the synthesis product concentration can be estimated by using appropriately designed interval observer. Based on this observer a robust-adaptive control strategy of a substrate inside the reactor is proposed. The performance of the proposed adaptive control strategy is illustrated by numerical simulations applied in the case of an anaerobic bioprocess for which kinetic dynamics are highly nonlinear, time-varying and incompletely known.**

Keywords- interval observer; adaptive control; wastewater treatment bioprocess; biomass estimation.

I. INTRODUCTION

In the case of biotechnological processes, a frequent and important challenge is finding adequate and reliable sensors to measure all important state variables of the plant. Nowadays, there are a number of on-line sensors able to provide state information, but they are very expensive and their maintenance is usually time consuming [1].

Furthermore, the biological systems contain living organisms that are not perfectly described by the physical laws, are strongly nonlinear, and they have poorly understood dynamics. This is one strong motivation for using robust methods for the control of this type of systems and for the estimation of the variables that are not measured [2]. By using control methods for the processes in the living organisms, the engineering motivation is to achieve better operational stability and production efficiency [3].

As biotechnological processes have gained an increasing importance in everyday life, a series of observers for nonlinear biological processes have been proposed, that can be chosen in accordance with the information available on the model that is being used [4].

In the last decade, the optimization of this kind of processes and mainly of wastewater treatment receives increased attention. The scientists are interested of designing new control strategies that guarantee a better process working and efficiency. However, these controllers often require high quality measurements or efficient state estimation procedures [10].

If a high gain observer is used for obtaining a good estimation of the internal state, the availability of a good model is necessary [3].

If in the parameters model is a large number of uncertainties, the best option is to use an interval observer, because this method provides an estimate of the quality [1].

Starting from the determination of the control law, we continued by considering all the states in the system known, in order to see how this is acting in open loop, and then in close loop.

Then, we considered that the process model was not completely known, and we proceeded to make estimations for unknown parameters.

The traditional control design involves complicated mathematical analysis and has difficulties in controlling highly nonlinear and time varying plants as well. A powerful tool for nonlinear controller design is the feedback linearization [5], but its use requires the complete knowledge of the process. In practice, there are many processes described by highly nonlinear dynamics; thus an accurate model for these processes is difficult to develop. Therefore, in recent years, great progress in development of adaptive and robust adaptive controllers has been noticed, due to their ability to compensate for both parametric uncertainties and process parameter variations. An important assumption in previous works on nonlinear adaptive control was the linear dependence on the unknown parameters [5].

Adaptive controllers have a strong advantage to the classical PID controller by the fact that can eliminate errors faster and with significantly reduce fluctuations. This advantage allows the process to have a higher profitability [9]. An important disadvantage of adaptive controllers is the requirement of a technical expertise for understanding how they can be fixed if they fail [9].

This paper is divided in five sections: in Section II, the mathematical model of a de-pollution process is described; Section III presents an adaptive control strategy, Section IV is dedicated to the formulation of an interval observer, and Section V concludes the paper.

II. MATHEMATICAL MODEL

Pollution can be defined by the modification of the physical, chemical and biological components which is damaging for the human beings and for the environment.

 ISBN: 978-1-61208-219-6

Pollution is obtained by adding pollutants in the environment [6].

In the last few years, the importance of biotechnology and of automatic control for de-pollution processes is permanently growing [6].

In this paper, we consider a process described by the following reaction [6]:

$$S \xrightarrow{r\leftarrow} X + P \tag{1}$$

In equation (1), the following symbols appear:

X – Biomass concentration [g/l]
S – Substrate concentration [g/l]
P – Synthesis product concentration [g/l]
r – reaction rate that is defined by the equation:

$$r = \mu(S,P)X$$

This model is a very simple one, because only one main bacterial population is considered.

This is a prototype of the de-pollution bioprocesses whose dynamic model is described by the equations [6].

$$\frac{dX}{dt} = \mu(S,P)X - DX_1 \tag{2}$$

$$\frac{dS}{dt} = -k_1\mu(S,P)X - D(S - S_{in}) \tag{3}$$

$$\frac{dP}{dt} = k_2\mu(S,P)X - DP \tag{4}$$

Beside the symbols from equation (1), in the system (2)-(4), there are also present the variables:

D – Dilution rate [h^{-1}]
μ – Specific growth rate [h^{-1}]
k_1, k_2 – maintain constant [dimensionless]
S_{in} – feed substrate concentration [g/l]

Specific growth rate is assumed to be of the following form [4]:

$$\mu(S,P) = \mu^* \frac{S}{K_M + S + S^2/K_I} \frac{P}{K_P + P}\left(1 - \frac{P}{P_L}\right) \tag{5}$$

The maximum specific growth rate is noted with μ^*; K_M is a notation for half the saturation constant associated with S, and with K_I the inhibition constant is noted [1].

The kinetic parameters necessary to calculate the specific grow rate have the following values [6]:

$$\mu^* = 0.53\,h^{-1};\quad K_M = 0.26\,g/l\quad K_I = 297\,g/l;$$
$$K_P = 7.77\,g/l\qquad P_L = 85.81\,g/l$$

The yield coefficients for the proposed model are [6]: $k_1 = 1.5$ and $k_2 = 1$.

The initial condition that are use for this paper are [6]: $X(0) = 0.37$, $S(0) = 90$, $P(0) = 6.1$, $S_{in}(0) = 100$.

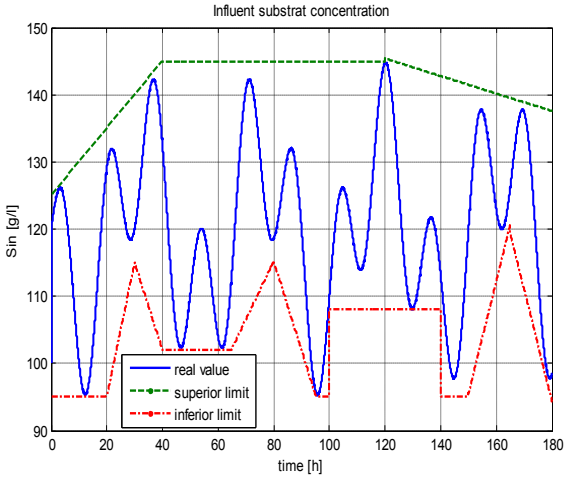


Figure 1. The evolution of feed substrate concentration S_{1in}.

In figure 1, a possible form for the real feed substrate is presented. Because we don't know anything about the real shape of feed substrate evolution, we can choose any form that is placed in a bounded domain ranges between the two limits: an inferior limit denoted by S_{in}^- and a superior limit denoted by S_{in}^+. Because of only this limits of the feed substrate concentration must to be known, the shape of real feed substrate can be completely random.

The real feed substrate concentration in Fig. 1 is also corrupted with an additive white noise with zero average (5% from their nominal values).

III. CONTROL STRATEGIES

A. Control Objective

The control objective consists in the adjustment of the system in order reduce the level of pollution in the wastewater. To be exact, considering that the process model (2)-(4) is incompletely known, its parameters are time varying and not all the states are available for the measurements, the control goal is to maintain the process at some operating points, which correspond to a minimal pollution rate [7].

We have chosen that the operating point is best to be kept around the point $S^* = 35$ g/l.

B. Exactly Linearizing Feedback Controller

First, we want to evaluate the ideal case where all the knowledge concerning the process (kinetics, yield coefficients and state variables) is available [14].

 ISBN: 978-1-61208-219-6

Let us consider a closed loop system whose dynamics are in accordance with a stable first order linear system, which is described by the equation below [15]:

$$\frac{d}{dt}\left(S^* - S\right) + \lambda_1\left(S^* - S\right) = 0 \tag{6}$$

Assuming $\lambda_1 = 0.5$, $\frac{dS^*}{dt} = 0$, we will have:

$$\frac{dS}{dt} = \lambda_1\left(S^* - S\right) \tag{7}$$

Replacing $\frac{dS}{dt}$ in (3) with (7), we will obtain:

$$\lambda_1\left(S^* - S\right) = -k_1\mu(S,P)X - D\left(S - S_{in}\right) \tag{8}$$

The control law, which is deduced from the above equation, is given by:

$$D = \frac{1}{S_{in} - S}\left(\lambda_1\left(S^* - S\right) + k_1\mu(S,P)X\right) \tag{9}$$

Because in real experiments some states are not available for on-line measurements, in Section IV, we will estimate the biomass concentration using an interval observer. This means that the biomass concentration is only defined by its superior and inferior bounds. Since the control law (9) depends on X this becomes an adaptive control law described by the following equation:

$$D = \frac{1}{S_{in} - S}\left(\lambda_1\left(S^* - S\right) + k_1\mu(S,P)\hat{X}\right) \tag{10}$$

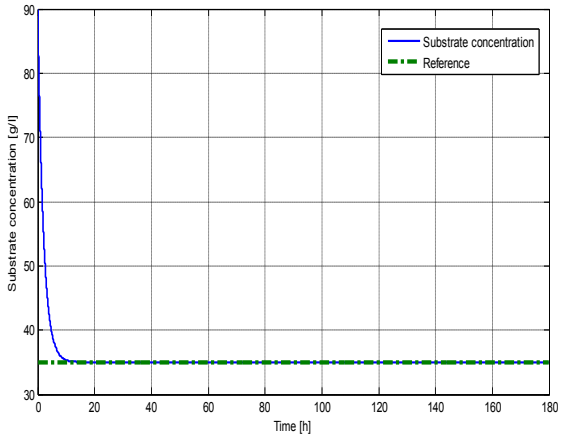


Figure 2. Substrate concentration when the reference is 35 (g/l).

In the equation (10), $\hat{X}$ stand for the arithmetical mean between the two estimated bounds of the biomass concentration.

Figure 2 presents the profile of controlled variable S when the reference is set to 35 g/l.

IV. INTERVAL OBSERVERS

Over the last ten years, a series of techniques were developed from which we mention: state estimation by means of convex sets, interval observers and bounded error estimators using interval analysis [2].

Interval observers provide state limits that will be estimated: the upper bound of the state vector provided by the upper observer, and a lower bound determined by the lower observer [3].

Interval observers were introduced in a general context by using the concept of framer, which is definite as a unique pair of estimates able to provide uncertain state limits, without taking into consideration any stability constraint [5].

Interval observers are based on positive differential systems and offer a way to deal with uncertainty in the system, when known bounds of the uncertain terms are available [5].

Note that in this process the feed substrate concentration S_{in} can be considered a disturbance whose exact values are not compulsory to be known, but, in the same time the values of S_{in} can vary between two known limits (an upper limit denoted with "+" and a lower limit denoted with "–"). Knowing these two limits we attempt to estimate all the other immeasurable state variables in model (3) by using all available information.

Therefore in the following we consider that the condition $S_{1in}^- \le S_{1in} \le S_{1in}^+$ is fulfilled.

To estimate the values of the biomass concentration X_1 and the synthesis product concentration P, first we introduce the following two additional variables [8]:

$$z_1 = k_1 X + S \tag{11}$$

$$z_2 = S + \frac{k_1}{k_2}P \tag{12}$$

The dynamics of z_i, $i = 1, 2$ deduced from the process model are expressed by the following linear stable equations [8]:

$$\begin{aligned} \frac{d\hat{z}_1^+}{dt} &= -D\left(\hat{z}_1^+ - S_{in}^+(t)\right) \\ \frac{d\hat{z}_1^-}{dt} &= -D\left(\hat{z}_1^- - S_{in}^-(t)\right) \end{aligned} \tag{13}$$

 ISBN: 978-1-61208-219-6

$$\frac{d\hat{z}_2^+}{dt} = -D\left(\hat{z}_2^+ - S_{in}^+(t)\right)$$
$$\frac{d\hat{z}_2^-}{dt} = -D\left(\hat{z}_2^- - S_{in}^-(t)\right) \quad (14)$$

that are independent of the process kinetics that could be completely unknown.

Because S_{in} is known only by its upper and lower limit, we can observe that in equations (13), (14), the additional values z_1, and z_2, can only be determined by their upper and lower limit values. For this reason, the real values of X_1 and P are located in an interval that is bound by these parameters' limits.

The on-line estimations of X_1 and P are given by the following equations:

$$\hat{X}^+ = \frac{1}{k_1}\left(\hat{z}_1^+ - S\right)$$
$$\hat{X}^- = \frac{1}{k_1}\left(\hat{z}_1^- - S\right) \quad (15)$$

$$\hat{P}^+ = \frac{1}{k_1}\left(\hat{z}_2^+ - S\right)$$
$$\hat{P}^- = \frac{1}{k_1}\left(\hat{z}_2^+ - S\right) \quad (16)$$

In equations (15) – (16) $\hat{X}_1$ and $\hat{P}$ represent the estimations of the biomass concentration X_1 and of the synthesis product concentration P, and the symbols "+" and "–" denoted respectively the upper and lower limits.

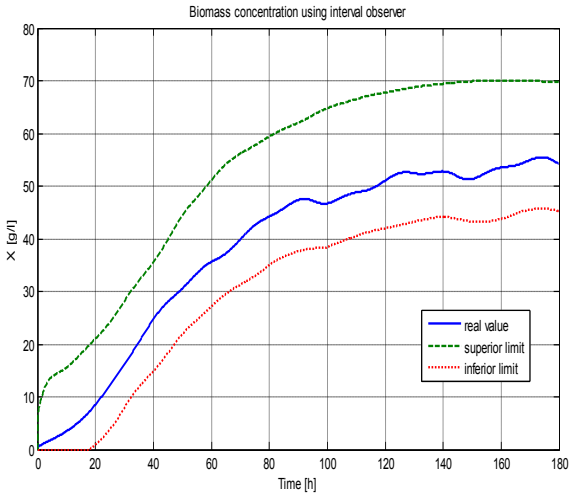


Figure 3. Profile of estimates of unknown biomass concentration versus its real value.

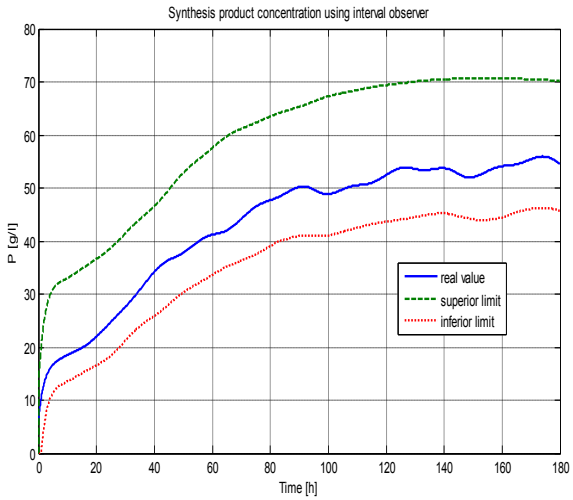


Figure 4. Profile of estimates of synthesis product concentration versus its real value.

The simulations have been carried out using the model comprised of the equations (2) – (4), considering that the process lasts for 180 hours and that the input concentrations are in a guaranteed interval.

The performance of designed interval observers (15)-(16) has been tested by performing extensive simulation experiments. The simulations were carried out by using the process model (2) – (4) under identical and realistic conditions.

In figure 3 and figure 4, the graphics marked with continuous line correspond to values of real concentration of S_{in}, as it should be known while the graphics marked with dotted and interrupt lines correspond respectively to known upper and lower limits of S_{in}.

Using different shapes of feed substrate concentration that are included between two known limits, it can be seen that the behaviour of the proposed interval observers are good, the values of the estimated biomass X_1 and of the synthesis product P remaining between the limits determined from equations (15) and (16).

V. CONCLUSION

In this paper, an interval observer for estimation of some uncertain state variables in an uncertain biotechnological process was designed. Using an interval observer only an upper and a lower bound of the biomass concentration and the synthesis product concentration can be estimated by using an appropriately interval observer. The designed observer was used in an adaptive control problem of a substrate inside a bioreactor.

The effectiveness and performance of the designed interval observers as well as of the adaptive control strategy were illustrated by numerical simulations applied in the case of an anaerobic bioprocess.

 ISBN: 978-1-61208-219-6

ACKNOWLEDGMENT

This work was supported partially by CNCSIS–UEFISCDI, Romania, project number PNII–IDEI 548/2008, and partially by the strategic grant POSDRU/88/1.5/S/50783, Project ID50783 (2009), co-financed by the European Social Fund – Investing in People, within the Sectorial Operational Programme Human Resources Development 2007–2013.

REFERENCES

[1] V.A. Gonzales – Estimation et commande robust non-lineaires des procedes biologiques de depollution des eaux uses. Application a la digestion anaerobie. These de l'Université de Perpignan, 2001

[2] M. Moisan, O. Bernard, and J.-L. Gouze - A high/low gain bundle of observers: application to the input estimation of abioreactor model, Proceedings of the 17th World Congress The International Federation of Automatic Control, Seoul, Korea, pp. 15547 – 15552, 2008

[3] M. Moisan and O. Bernard - Interval observers for non monotone systems. Application to bioprocess models, in proceedings of the 16th IFAC world conference, Prague, 2005.

[4] E. Petre, C. Marin, and D. Selisteanu – Adaptive control strategies for a class of recycled depollution bioprocesses, CEAI, Vol. 7, No. 2, pp. 25—33, 2005

[5] S. Sastry and A. Isidori - Adaptive control of linearizable systems. IEEE Trans. Autom. Control, 34, 11, pp. 1123—1131, 1989

[6] E. Petre and D. Selisteanu – Modelling and Identification of depollution bioprocesses, Ed. Universitaria Craiova, 2005

[7] D. Sendrescu, E. Petre, D. Popescu, and M. Roman, Neural Network Model Predictive Control of a Wastewater Treatment Bioprocess, The 3th International. Conference on Intelligent Decision Technologies – IDT 2011, Piraeus, Greece, pp. 191-200, 2011

[8] M. Moisan - Synthèse d'Observateurs par Intervalles pour des Systèmes Biologiques Mal Connus, These de l'Université de Nice Sophia – Antipolis, 2007

[9] V.J. VanDoren (Ed.), Techniques for Adaptive Control, New York: Butterworth-Heinmann (Elsevier Science), 2002

[10] M. Moisan, O. Bernard, and J.L. Gouze. Near optimal interval observers bundle for uncertain bioreactors. Automatica, 41, pp. 291–295, 2009.

 ISBN: 978-1-61208-219-6

An Adaptive Multi-agent System for Ambient Assisted Living

Antonio Andriatrimoson, Nadia Abchiche-Mimouni, Etienne Colle and Simon Galerne
IBISC laboratory, Evry Val d'Essonne University
Evry, France
{*tsiory.andriatrimoson, nadia.abchiche, etienne.colle, simon.galerne*}*@ibisc.univ-evry.fr*

***Abstract*—The work presented in this paper is focused on the design and the implementation of an adaptive framework for ambient assisted living applications. The challenge is to provide an approach able to deal with a dynamic environment in order to provide an adequate service to the person. The evolution of the intrusion level of the system based on the degree of urgency and the availability of different communication devices that constitute the environment are particularly targeted. The results obtained with the coalition-based multi-agents system are promising and reflects these constraints.**

***Keywords*-adaptiveness; multi-agent systems; cooperation; coalition formation.**

I. INTRODUCTION

Software increasingly has to deal with ubiquity, so that it can apply a certain degree of intelligence. Ambient assistive robotics can be defined as an extension of ambient intelligence which integrates a robot and its embedded sensors. The interaction among the components in such systems is fundamental.

The addressed problem here concerns the design of an ambient assistive living framework that takes advantage of an ambient environment: a robot cooperating with a network of communicating objects present in the person's home. The aim is to provide a service to an elderly or a sick person.

A multi-agent system (MAS) reifies the sensors and the mobile and autonomous robot, allowing the cooperation among the agents by means of adaptation features. Coalitions are formed in adaptive way as it will be described in Section IV.

The next section details the context application and describes a particular usage scenario. Section III includes a brief overview of existing ambient assistive living systems and argues for a new one based on adaptive coalition-based MAS. The designed system Coalaa is described in details in Section IV. First evaluations and analysis of Coalaa are presented in Section V. Finally, in Section VI we draw some conclusions and introduce future works.

II. THE PROBLEM DESCRIPTION

Ambient Assisted Living (AAL) constitutes a fundamental research domain. It refers to intelligent systems of assistance for a better, healthier and safer life in the preferred living environment and covers concepts, products and services that interlink and improve new technologies and the social environment, with a focus on older people. A panorama of European projects can be found in ([1]). Our specific context is to assist a person in loss of autonomy at home. It concerns either the elderly or people with specific disabilities. Maintaining such people at home is not only beneficial to their psychological condition, but helps reduce the costs of hospitalizations.

House is equipped with a network of communicating objects (CO) such as sensors or actuators for home automation. A complete telecare application for remote monitoring of patients at home, including a wireless monitoring portable device held by the patient, is added for detecting alarming situations

The context application is essential in this work. So, a usage scenario is described in details so as to illustrate the different application challenges and the scientific issue addressed in this paper which is implementing adaptiveness.

A. A scenario description

The scenario consists in a variety of situations where an alarm has occurred (The scenario has been determined in cooperation with the remote monitoring center SAMU-92, which depends on Public Paris Hospital).

An alarm can be triggered by a device worn by the person or the sensor network of the ambient environment. The robot, thanks to its ability to move, helps to confirm and evaluate the severity of the alarm by cooperating with the CO.

The robot begins by searching the person and then provides an audiovisual contact with a distant caregiver. That way, the distant caregiver is able to remove the doubt of a false alarm, to make clear the diagnosis and to choose the best answer to the alarming situation. It is important to note that the embedded device monitors the physiological parameters and the activity of the person. The originality of the proposed approach is that the robot tries to take advantage of ubiquity. The robot autonomy is obtained by a close interaction between the robot and the ambient environment (AE). So, the services the robot can bring to the user are directly related to the effectiveness of the robot mobility in the environment. Before providing a service to the person, the robot has to locate itself by interacting with the AE. In these scenarios an ethical dimension, named level intrusion of the system, has been introduced to preserve the

 ISBN: 978-1-61208-219-6

privacy of the person. The level of intrusion of the system is defined according to the degree of freedom of the system regarding to its actions. For instance: maximal distance allowed between the robot and the person, activating a camera, switch on a light and so on. The level of intrusion of the system is supposed to be minimal except in a case of an emergency.

B. Robot localization task

Using a robotic assistant for the task rather than a simple set of fix cameras in all rooms is an advantage in two cases: i) the assistance is only needed for a limited period such as convalescence period or ii) the residence is composed of too many rooms for example nursing home. Another advantage is that the quality of the image and the sound is better. The part of the robot is to autonomously move to the person in case of an alarm and then provide an audiovisual contact with a distant surveillance center.

Figure 1 shows a robot in the person's home; the patient has fallen. To move towards her/him and to guide its camera to the remote caregiver, the robot has to be located first. A visual contact will help the remote caregiver to perform a correct diagnosis of the situation.

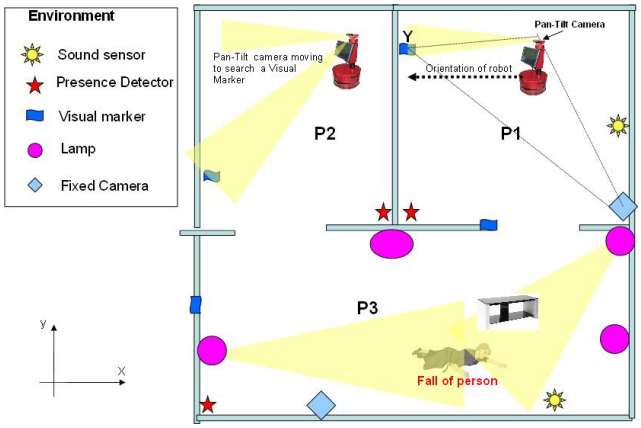


Figure 1. A person falls scenario

If the robot is located at P1 position, then its mobile camera can identify the visual marker Y. With further information from a fixed camera environment, the robot manages to locate itself by a mean of an adequate localization algorithm. The direction taken by its mobile camera that detected a visual marker also allows the robot to know its orientation relative to a fixed reference in the environment. This information can also be inferred from previous values using odometry on the one hand and its linear and angular speeds on the other hand. It is thus easy and straightforward to identify and understand that the more information you have the better the accuracy of the location of the robot is.

If the robot is in P2 position, it has no marker on its visual field and has no element enabling it to locate itself. It then uses two different strategies to find a visual marker. Either its moves randomly or turns its pan-tilt camera. In two cases, it is necessary that the intrusion level of the system permits it. It can also query the detectors of presence to learn about the place in which it has been seen lately. In the case of several conflicting reports, it will be decided according to the data freshness criteria, or according to the consistency with the data criteria already available thanks to the sensors of the robot.

This simple scenario shows that robot localization is a complex task and there is no evidence for an approach that could be able to choose the relevant interactions between the robot and the A.E. The difficulty lies in choosing the most relevant criterion to be considered first: is it the closest CO, the most accurate and or least intrusive? The problem analysis suggests that depending on the context, the criterion to consider is different. As the context itself is dynamic and difficult to predict, a centralized algorithmic solution is to be excluded. What is required is an approach that can adapt the selection and the use of criteria based on the context and the choice of a level of intrusion aligned with the level of urgency. Adaptive systems ([2]) are known to meet this requirement. More precisely, adaptation features are inherent to MAS. So, our approach exploits the MAS adaptiveness potential to design a distributed system to deal, in a dynamic way, with scenarios such as the one described above. The adaptiveness is also needed to deal with dynamic addition and suppression of sensors. While the purpose of the paper is not to describe the localization algorithm but a selection mechanism of the agents participating to this task, it is not necessary to explain the robot localization.

III. State of the art

Before addressing a state of the art in the MAS domain, a brief overview of existing ambient assistive living approaches is given.

A. Ambient assistive living existing approaches

In the context of ambient intelligence, the communicating objects of the AE play a "facilitator" role in helping the robot in the Ambient assistive living.

Conversely, sensors and robots can be seen as communicating objects which are used by services to the person in loss of autonomy. Several projects have been interested in combining home automation, pervasive sensors and robotics, for the safety of the patient at home.

The IDorm project ([3]) is designed to assess an ambient environment composed of three categories of communicating objects: static objects associated with the building, a robot and mobile devices. IDorm architecture consists of a MAS that manages the operations of all the environment sensors and the robot. The sensors are controlled by an agent and the robot by another one. The sensor agent receives the different measures from sensors and controls actuators which are

 ISBN: 978-1-61208-219-6

linked to sensors like a pan-tilt camera. The robot agent acts as a data server and coordinates exchanges of information between the user and the robot. It controls the navigation of the robot by combining different functions such as the obstacle avoidance or the search for targets.

The CARE ([4]) project is a Research and Development activity running under the Ambient Assisted Living Joint Program, which is co-funded by several European countries. Its main objective is fall detection and person monitoring at home by Smart camera. As part of this project, algorithms essentially based on a biologically-inspired neuromorphic vision sensor for fall detection have been developed. The system aims to define a level of reliable supervision by avoiding as much as possible interactions with the person in her/his own home.

ProAssist4Life ([5]) is a German project of situation-of-helplessness detection System for elderly. This project consists in developing an unobtrusive system that provides permanent companionship to elderly people living in single households or in retirement facilities. Multisensory nodes mounted on the ceiling of a room register an individual's movements. One multisensory node contains six motion sensors, one brightness sensor, and one oxygen sensor. According to data provided by various physiological sensors, the system is based on a predictive approache based on finite state automata modeling the previous activities of the patient.

Another project developed at the University of Camerino is named ACTIVAge ([6]). In order to keep people at home also, this project aims to provide services and teleservices based on the context. The system consists of an adaptive planning solver based webservices orchestation and choreography with decision making algorithms. A knowledge base is used to model persistent data of the ambient environment.

In each of these projects, the authors seek to design a system to avoid interfering with the patient at home. Ethical dimension is still much debated in the field of ambient assisted living, this constraint is managed by the projects mentioned above by discrete sensor systems. Although the last described project pretends dealing with adaptiveness, this concept remains a major challenge in ambient assistive applications.

The work presented in this paper is focused on implementing adaptiveness while designing several application aspects. The evolution of the inconvenience (intrusion level of the system) based on the degree of urgency and the availability of different communication devices that constitute the environment are particularly targeted. The coalition-based MAS presented in this paper reflects this constraint.

The purpose of the paper is to describe a selection mechanism of the agents participating to the localization task, so localization algorithm is not presented in details.

B. Coalition-based protocols

The principle of coalition aims at temporarily putting together several agents for reaching a common goal. Several woks have illustrated the relevance of coalition-based approaches for adaptiveness ([7][8][9]). The methods are various: either incremental or random or centralized. But, all of them proceed in two stages: (1) the formation of agent coalitions according to their ability to be involved in achieving a goal and (2) the negotiation stage between the coalitions in order to choose the one that provides the closest solution to the goal. The interests of the coalition-based formation protocols are the flexibility with which coalitions are formed and straightforwardness of the coalition formation process itself. The coalitions can get rid of dynamically reorganize with local and simple rules defined in the agents.

IV. Coalitions for Ambient Assisted living

Coalaa (Coalitions for Ambient Assisted living applications) is a MAS ([10][11][12]) based on coalitions formation protocol. Each agent encapsulates a CO. It decides in a local and proactive way how to contribute to the required service to the person. In fact, we have introduced a more general notion than a service, that we have called an effect. An effect can be a particular lighting at a precise place of the residence or the localization of a robot. The MAS configures itself for providing a solution according to the availability of the CO and the respect of criteria. The adaptation to the context is inherent to the multi-agent modeling, strengthened by coalitions and negotiation mechanisms. Note that the goal is not to find the optimal solution but a solution close enough to the required effect.

In our coalition formation protocol, the obligation to respect the result and an intrusion level depending on the urgency of the situation, are the most important considered criteria. They are also used during the reorganization of the agents for searching for a desired effect. The obligation result criteria is used in priority while the level of intrusion is modified only if needed, i.e., to acquire new data and thus to activate the sensors (ex. tilt-camera) likely to cause discomfort to the person.

A. Knowledge modeling

An effect is modeled in the form of a triple σ = <t, c, f>.

- t $\in$ T, T is a set of tasks: localization of a robot or a person, lighting, cognitive stimulation.
- c $\subset$ C, C is a set of criteria: precision, efficiency, time constraint, neighborhood.
- f $\subset F$, F is a set of influencing factors: intrusion level, urgency degree.

The criteria are assigned by the designer (programer) of the system in a static way, while the influencing factors are dynamically fine-tuned by the end-user.

 ISBN: 978-1-61208-219-6

B. Agents environment

The ambient agents operate in an ambient environment consisting of habitat model within which the patient and the robot are together. They argue according to the different measures and relevant information that smart objects provide.

1) Ontology: Information handled by the system is classified into two types. This so-called persistent information, related to the application domain, puts together data about the structure of the residence and the features of the CO. The second type concerns volatile data mainly the measures provided by the sensors and the orders sent to actuators. The information types are handled differently. The volatile data are distributed in each agent, while persistent data are stored in an ontology named AA (Ambient Assistance) ([13][14]). The AA ontology contains four categories of information related to the application domain: The Home category for defining the structure of the environment, the CO category for knowing their characteristics and their operating mode, the User category for defining the user profile and the Task category that puts together the tasks and services that the system is able to achieve. These categories define the four concepts of the ontology. Our system needs to set up links between members of the same concept such as a topological relationship between two parts of the residence. Links are also needed between members of different concepts. For example, to process a measure provided by a sensor, the system has to locate the sensor in the residence. These links are referred to as ontological properties. We have defined three types of properties: relationship, use and attribute. The ontological property relationship defines a logical relationship, generally of ownership, which links concept members between each other. The ontological property use defines the function of an object. The ontological property attribute refers to the features of a concept or a concept of an individual member of the ontology. It specifies the operating mode of the object, for example, a camera can be used to perform the localization task.

In this ontology, a property named topological distance is defined as the number of hops between two instances. The hops are relations as defined above in the ontology. If the structure of the ontology is defined by a graph, the topological distance is the number of nodes which separate two individuals minus one. This topological distance is used by agents of the MAS to determine their neighborhood during the coalition formation.

This knowledge base is complemented by the dynamic information from the ambient environment through the gateway.

2) Gateway: It is a module for the standardization of information exchanged between the ambient environment the MAS. Its role is to make the agents manipulating the common information format.This standardization is necessary because of the heterogeneity of protocols from different manufacturers. Thus, the MAS receives and acts on the ambient environment through the gateway without worrying about the format of the collected data.

C. Agent internal architecture

Figure 2 represents the internal architecture of an ambient agent. The decision making module takes in charge the agent adaption and reactivity by using three main parameters that are neighborhood, history, and ability. The neighborhood sets the list of agents that are close to this agent at a given time, according to the topological distance.

The history stores previous perceived information which come from the sensors. This is a simple succession of perceived data which helps to consider the timescale during the process of coalitions formation.

At last, the ability identifies the skills of the agent which are directly related to the encapsulated CO.

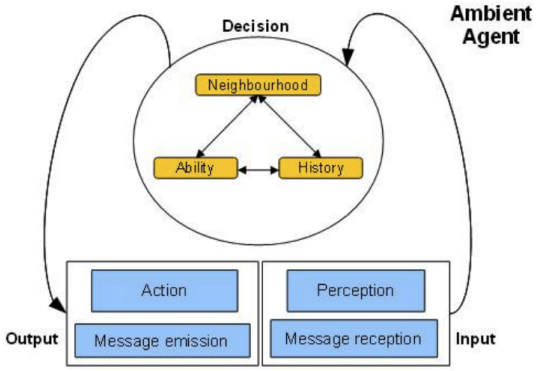


Figure 2. Agent internal architecture

D. Agent behaviors

In the process of the coalition formation, an agent may be either initiator or candidate. Any agent whose ability can partially meet the desired effect can be a coalition initiator. The initiator exchanges messages with other agents, potential members of the coalition, called candidate agents. The Protocol is based on exchanges of messages between the initiator agent and candidate agents. As soon as the overall ability of the coalition is close to the desired effect, the initiator agent is pending the negotiation phase. At the end of the coalition formations, each initiator agent that is the referent of the coalition is negotiating with other initiators agents to choose the winning coalition. The coalition whose ability is the closest to the desired effect is the winning coalition.

The concept of ability is general. In the localization application example, it is instantiated by the measures precision.

The principle is simple. Each initiator agent sends a message that contains the ability obtained by its coalition. On receipt of this message, each initiator agent compares

 ISBN: 978-1-61208-219-6

the ability of the coalition it received to its own one. If its ability is lower than that received, the coalition will be no more considered, otherwise, it is a winning coalition up to receiving a new message. Apart from the desired effect, the formation of coalitions uses other criteria such as the topological neighborhood to reduce the response time or the obsolescence of a measure when the desired effect depends on sensor data. Thus, the first step is the identification of candidate neighboors according to its own location in the environment (defined by the topological distance) and the desired effect. The aim of this strategy is to respond in the shortest time to the desired effect by forming coalitions. For that purpose, the first selection criteria considered is the topological distance. Once all candidate agents are known, each initiating agent continues the selection of candidates based on the recent measures criteria. When no coalition is able to meet the desired effect, a new search for a successful coalition is restarted after having relaxed the constraints on certain criteria. Indeed, it is possible to increase the level of intrusion of the system despite of the tranquility of the person at home. This authorization to increase the level of intrusion allows, for example, to operate a pan-tilt camera of the robot to acquire new measures and restart the process by finding a winning coalition.

The MAS protocol is defined as a set of rules that ambient agents follow to find out a solution. The protocol of coalition formation is composed of two distinct steps. The first step consists in forming coalitions of agents according to their ability. The second step is a negotiation and refining phase so that the best one, in satisfying the desired effect, is chosen.

In summary, after initialization, these exchanges follow three main actions:

1) Formation of all possible coalitions for each referent.
2) Selection of the best coalition according to the coalition precision.
3) Deployment of the winning coalition.

To make decisions and follow the protocol, each agent executes the appropriate behavior and starts in a state corresponding to the behavior adopted.

Figure 3 shows the state transition diagram of the behavior of an ambient agent. Each ambient agent includes six parallel and cyclic behaviors. The Baseline behavior represents the minimum treatment of an agent. Upon receipt of a frame from the environment (by the mean of the gateway), the agent must recover the sensor ID associated with it, therefore it can access the ontology and update ability. InitCoal Behavior, AcceptCoal Behavior, ACKCoal Behavior and InitNegociation Behavior include the process of coalition formation and negotiation. For the formation of coalitions, the first behavior to be executed is sending InitCoal following receipt of a InitEffect. Running an InitCoal behavior consists in sending a message, containing the ability of the agent, to the neighborhood agent. All agents which receive this message accept or refuse to be part of the coalition. Agents which accept must then execute the AcceptCoal Behavior and then send an acceptance message or refusal message. Initiators which receive an acceptance reply with a confirmation. Finally, the EndNegociation Behavior runs when a winning coalition has emerged. This is a behavior that allows the deployment of the coalition.

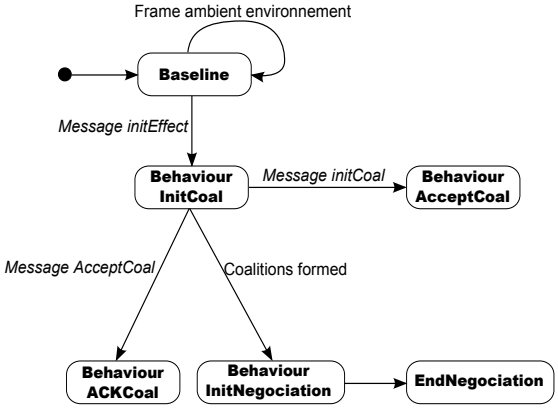


Figure 3. Behaviors of an agent

E. Agents interactions

For the formation of the coalitions, two main types of messages are defined: Request message and Response messages. The exchanged messages semantic is based on speech act theory, introduced by John Searle ([15]), allowing the agents to assign the messages a semantic by defining a message a subtype.

1) Initialization: Initialization messages subtype is used in two situations: by the Interface Agent (AI) to send an effect to achieve (InitEffect) to the agents of the system and, the initiator agents after all coalitions have been formed so that it is possible to initiate the negotiation (InitNegociation).

2) Coalition: A Coalition message type is sent in response to the reception of a desired effect.

3) Acknowledgement: A confirmation message (ACK-Coal) or a refuse message (RefuseCoal) is an Acknowledgement message subtype.

4) Reaction: This subtype includes two main messages that are AcceptCoal and ArgNeg. AcceptCoal is a message Reaction subtype that is sent by an agent when accepting an InitCoal proposal. The second message Reaction subtype is ArgNeg that is sent by an agent to respond to a request for negotiation.

Each message type contains the ability of the sending agent while forming the coalitions, and the ability of the coalition during the negotiation step.

F. Agent genesis

The initialization step of the MAS is performed by a particular initialization module. It is to trigger a behavior

 ISBN: 978-1-61208-219-6

that scans the environment of each agent and creates the agents. Each created agent is initialized by loading locally, a data set from the ontology and information from the physical environment (the gate).

G. Robot localization scenario

In this scenario, three sensors of the environment are used: a robot pan-tilt camera, a fixed camera and a presence detection sensor. These three communicating objects are encapsulated by three respective ambient agents: a Presence Detector Agent (APD), a Fixed Camera Agent (AFC) and a Pan-Tilt Camera Agent (APTC). Visual markers like Datamatrix are associated with each camera.

Figure 4 shows a sequence diagram of the different agents that are involved in the scenario already described in Section II.A.

Following the fall of the patient, a request for a localization effect is generated in the form of a triple σ = <t, c, f> (cf. Section IV.A). t is the localization task which matches with the localization effect, c matches with a singleton containing the precision criterion needed for the localization task and f matches with a set containing two influencing factors that are: the intrusion level and level of urgency. In the considered scenario, we have considered a precision equal to 0.1, a level of urgency equals to 3(three levels of urgency are considered: low=1, medium=2, high=3) and an intrusion level initialized to 0 (the less intrusion level). So, the tripe becomes: $< Locate, \{0.1\}, \{3, 0\} >$.

The Interface agent (AI) has received the desired effect and then broadcasts the request InitCoal ($< Locate, \{0.1\}, \{3, 0\} >$) to all the agents of the MAS. Each agent which received the desired effect checks its ability. As all sensors in the environment have a precision that is not better than the desired effect, each agent initiates a coalition with immediate neighborhood. In this figure, only interactions with APD agent are shown. Assuming that all agents are topologically close, APD broadcast a coalition formation request by sending an InitCoal message. Each agent receiving the initialization message checks if its ability is adequate with the request of coalition formation.

If yes, it sends an acceptance message labelled AcceptCoal to be a candidate. Such a message contains the precision of the agent.

APD adds progressively answer acceptance, and accumulates the abilities which are the precision in the considered localization task.

By this way, it calculates the overall ability of the coalition until it reaches that of the desired effect.

Then, it sends ACKCoal acceptance to confirm the membership of the candidate to the formed coalition.

V. CONTRIBUTIONS AND RESULTS/OUTCOMES

The results are obtained in a real environment composed of heterogeneous sensors and markers. The platform includes several sensors obtained on the market and dedicated

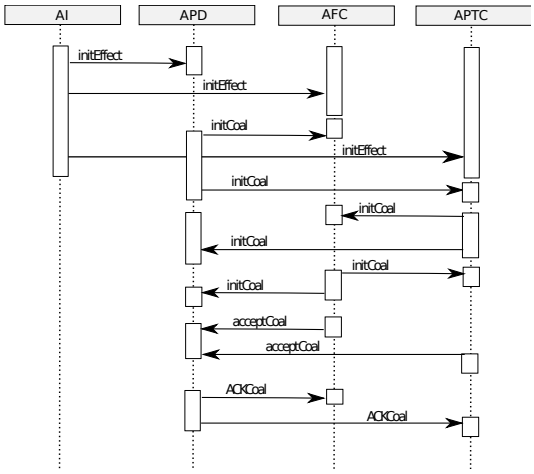


Figure 4. Sequence diagram

sensors developed in the laboratory. The environment is composed of a room equipped with a set of sensors and the robot with its own sensors. The simple localization presented scenario has been chosen because the principle is to use the orientation measurement of various sensors or markers. These can provide localization information to obtain the localization of the robot in its environment using real-time data either from the robot on-board sensors or from the sensors in the environment.

Coalaa has been implemented using a multi-agents system plateform: Jade ([16]). Jade provides generic behaviors which facilitates controlling the execution of the agents.

Adaptation in our system is observed at three levels; (1) computational level: during the coalition formation process, (2) functional and methodological level: while service modeling, and (3) ethical level: intrusion level of the system which is integrated in the behavior of the system.

A. Computational adaptiveness

To validate the protocol used in Coalaa, a comparison to a well known protocol which is the Contract Net Protocol (CNP) has been performed. The CNP was the first approach used in MAS to solve the problem of tasks allocation. Proposed by Smith in 1980 ([17]), it is based on an organizational metaphor. The agents coordinate their work based on building contracts. There are two types of agents, a manager agent and contracting agents. The contractor agent must complete a task proposed by the manager. The manager breaks down each task into several subtasks, and then announces each subtask to a network of agents by sending a proposal. Agents contractors which have adequate resources respond by sending their submission. The manager agent analyses all received bids and based on the result of this analysis assigns the task to the best contractors. The contractors commit with the manager to perform the

 ISBN: 978-1-61208-219-6

assigned subtask.

The CNP and the Coalaa protocols have been tested with a dozen scenari using in each scenario, different values for the criteria. Each scenario has been executed with both protocols. The showed results represent an average of the results of the scenari.

Evaluations have been performed on a MAS whose cardinality varies. The results are broken down into three categories:

1) The number of formed coalitions (see Figure 5),
2) The comparison of the response time (see Figure 6),
3) The number of exchanged messages (see Figure 7).

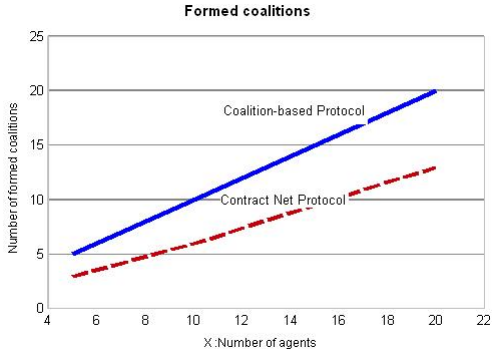


Figure 5. Formed coaltions

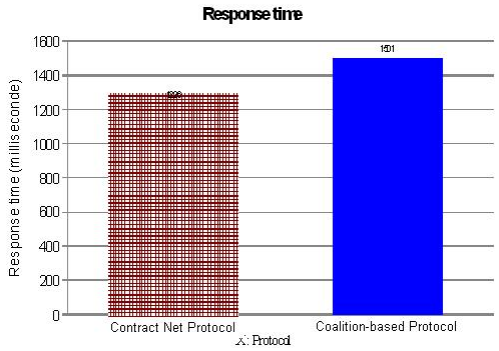


Figure 6. Response time

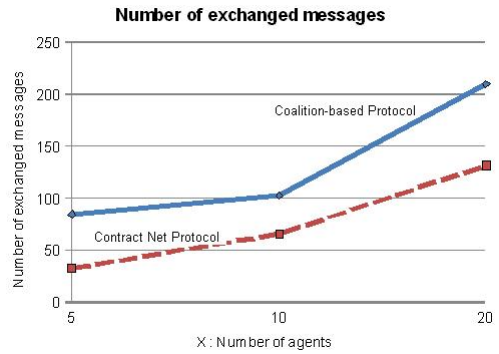


Figure 7. Exchanged messages

Figure 5 shows the number of formed coalitions depending on the number of agents present in the MAS. The preferred strategy in our approach is to obtain a maximum number of coalitions that meet the selection criteria. The goal is to maximize the number of solutions to meet the request to increase the chances of securing a result. The number of coalitions is always equal to the number of initiators. In terms of the number of formed coalitions, the Contract Net protocol is less efficient than Coalaa protocol. The response times is compared (see Figure 6). This time corresponds to the time spent in calculating the coalitions, including the message exchanges.

The fact that the number of coalitions that the CNP can form is lower than the number of initiators has a direct effect on the response time. It also impacts the number of exchanged messages represented by Figure 7.

The curve representing the number of exchanged messages follows the same rate for the two protocols. However, Coalaa shows a higher number of exchanged messages. Unlike the CNP, Coalaa avoids system crashes, by a progressive coalition formation which in contrast increases the number of exchanged messages. In terms of performances (time response and number of exchanged messages) measures Coalaa and CNP are almost similar; CNP is slightly better in terms of response time. But in terms of obtained results Coalaa is better. Indeed, a failure can be catastrophic and thus the few milliseconds delay in the response time may be insignificant, if success to complete the task is assured.

This is explained by the fact that Coalaa continues to reorganize itself until finding a solution (even with deteriorated criteria), while with CNP, the system can fail and do not offer a solution.

B. Methodological and functional adaptiveness

The genesis of the MAS is done automatically. In spite of the fact that this has not been detailed in this paper, this is very important feature of the system. In fact, modifying the ambient environment, by adding or suppressing CO, automatically updates the ontology Habitat and triggers automatic MAS reconfiguration. In case of such modifications, the user does not need to do any specification to make the system adapting its architecture to AE dynamic updating. This ability of the system is qualified by methodological adaptiveness. We refer to functional adaptiveness when dealing with services that the system can offer to the user. The description of the ability of the CO used by the agents to construct services according to the ”effect description” is included in the ”task” ontology part. This allows the agents to perform an automatic detection of their ability to perform an effect.

 ISBN: 978-1-61208-219-6

C. Ethical adaptiveness

An original specificity of our system is its dealing with an ethical dimension, that is the level of intrusion of the system. In fact, the system is able to adapt the intrusion of the robot, the CO and the embedded software according to the urgency of the situation and be allowed to cause discomfort for the person or its entourage only if needed.

VI. Conclusion and future works

An adaptive approach has been presented for an assistive ambient alarm detection by implementing the Coalaa system. Coalaa is a coalition-based multi-agent system in which the adaptiveness is considered from the computational, the methodological and the ethical points of view. The feasibility of this approach has been demonstrated on a usage scenario to remove the doubt of a false alarm. The first results illustrated with robot localization are promising. Moreover, comparing our protocol to the contract-net protocol has shown that even more time is spent with Coalaa, the number of the solutions is greater. We think that the speed of Coalaa can be improved by revising the way of choosing the criteria priority. Indeed, in spite of conclusive results, several improvements of Coalaa are under consideration.
Current work concern the validation of the system with a great data size. The generation of statistical distributions of data will provide more more meaningful results. Another a work in progress is to implement more flexible way to calculate the cardinality of the coalitions. This could be done by the agents by evaluating their behavior and self-adapt for improving the overall model of criteria evaluation ([18]). At a short-term perspective, we plan to apply our approach to other services such as cognitive stimulation and detecting of the person activity. At a long-term perspective, we will propose to wrap an agent in each communicating object, so that no time is spent to acquire information from a gate and apply Coalaa as a solution to optimally deploy the sensors in the houses.

References

[1] J. Chailloux, "Special theme: Ambient assisted living," October 2011.

[2] A. Eduardo, D. Kudenko, and D. Kazakov, *Adaptation and Multi-Agent Learning*. Springer-Verlag Heidelberg, 2003.

[3] P. Remahnino, H. Hagras, N. Monekoss, and S. Velastin, "Ambient intelligence a gentle introduction," *Ambient Intelligence A Novel Paradigm*, 2006.

[4] A. N. Belbachir, F. Vajda, T. Lundn, and E. Schoitch, "Smart caring cameras for safe and independent living of the elderly: A non-wearable technology and service for fall detection," *ERCIM NEWS 87*, october 2011.

[5] D. Rombach, H. Storf, and T. Kleinberger, "Situation-of-helplessness detection system for senior citizens," October 2011, pp. 32–33.

[6] F. Corradini, E. Merelli, D.-R. Cacciagrano, R. Culmone, L. Tesei, and al., "Activage: proactive and self-adaptive social sensor network for ageing people," *ERCIM NEWS 87*, pp. 36–37, October 2011.

[7] M. Sims, C. Goldman, and V. Lesser, "Selforganization through bottom-up coalition formation," in *the 2nd AAMAS*, 2003.

[8] T. Scully, M. Madden, and G. Lyons, "Coalition calculation in a dynamic agent environment," in *the 21st ICML*, 2004.

[9] L.-K. Soh and C. Tsatsoulis, "Reflective negotiating agents for real-time multisensor target tracking," in *IJCAI'01*, 2001.

[10] M. N. Huhns, *Distributed Artificial Intelligence*. Pitman, 1987.

[11] M. N. Huhns and M. P. Singh, *Readings in Agents*. Morgan Kaufmann, 1997.

[12] M. Wooldridge and N. R. Jennings, "Agent theories, architectures, and languages: a survey," in *Intelligent Agents*, Wooldridge and J. Eds, Eds., Berlin: Springer-Verlag, 2009, pp. 1–22.

[13] A. Kivela and E. Hyvonen, "Ontological theories for the semantic web," in *Semantic Web Kick-Off in Finland*, May 2002, pp. 111–136.

[14] R. Arnand, E. M. Robert, H. C. Roy, and M. M. Dennis, "Use of ontologies in a pervasive computing environment," *Knowledge Engineering Review 18-3*, p. 209220, 2003.

[15] J. Searle, "Speech acts. an essay in the philosophy of language," *Cambridge University Press*, 1969.

[16] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley, February 2007.

[17] R. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver," in *IEEE Transactions on computers*, 1980, pp. 1104–1113.

[18] F. Klugl and C. Bernon, "Self-adaptive agents for debugging multi-agent simulations," in *ADAPTIVE 2011*, 2011, pp. 79–84.

 ISBN: 978-1-61208-219-6

Using Role-Based Composition to Support Unanticipated, Dynamic Adaptation - Smart Application Grids

Christian Piechnick, Sebastian Richly, Sebastian Götz, Claas Wilke and Uwe Aßmann
Technische Universität Dresden
Software Engineering Group
01062 Dresden, Germany
Email: {christian.piechnick, sebastian.richly, sebastian.goetz1, claas.wilke, uwe.assmann}@tu-dresden.de

***Abstract*—Due to the wide acceptance and distribution of mobile devices, it has become increasingly important that an application is able to adapt to a changing environment. This implies the necessity to integrate varying functionality at runtime being activated depending on the current context. A common approach is to foresee and model all possible influencing factors and to integrate the required software building blocks in advance. But, due to the constant change of the environment, as described by Lehman's laws, it is impossible to anticipate all future situations. Hence, modeling the entire adaptation process at design time prohibits the adaptation to unanticipated scenarios and, thus, is likely to lead to the malfunctioning of the adaptive application in the future. In this paper we focus on unanticipated, *dynamic self-variation* of applications (i.e., without a central coordinator) and propose a *role-based composition system* that enables the adjustment of the structure and functionality of software-objects in a fine-grained manner. Systems following our proposed approach form a Smart Application Grid (SMAG). The SMAGs-Approach is putting emphasis on dynamic collaborations between components within an application and between several different software systems. Therefore, *role-modeling* is used to model and perform dynamic variation of applications at runtime, whereby roles are stored in central repositories. This allows the integration of previously unknown software-building-blocks and the dynamic adaptation to situations that were not foreseen.**

Keywords-Dynamic Variation; Unanticipated Adaptation; Role-Modeling; Composition; Repository.

I. INTRODUCTION

Software has become ubiquitous and plays an essential role in almost every area of life, ranging from small personal tasks in everyday life to coordination and partial autonomous control of global economic processes. Our current world is characterized by high dynamics and the pressure of constantly adapting to a changing environment. This, however, leads to the requirement that software applications have to adjust themselves to changing requirements just as quickly. These adaptation processes, which in most cases cannot be predicted due to the complexity of open world scenarios, traditionally rely on static software development life cycles. As part of the establishment and widespread acceptance of mobile devices, their applications are becoming a fundamental part of daily life. This further increases the pressure to automatically adapt an application at runtime to its constantly changing application context (e.g., location, time, task and collaboration partners) [17]. In complex scenarios, the possible values and changes in external conditions are usually not predictable. This makes it impossible to plan all potential adaptation operations at development time.

Unlike conventional product enhancements or patches within the software development process, dynamic context adaptations are small, spontaneous and fine-grained runtime changes in both application structure and functionality. In large, monolithic applications, where only large building blocks can be exchanged, adaptation processes can become very complex and costly. In contrast, dynamic networks of small cooperating applications that are prepared on variability in advance are more suitable for runtime adaptation. In this way, the timespan from the occurrence of a changed requirement to an adequate adaptation of the application is significantly reduced and costly product development cycles are avoided. The finer-grained the interchangeable elements of the application are, the better the impact of change operations on the integrity of the entire system can be estimated. This is because the elements relate to specific functions, which in the ideal case have only limited effect on the correctness of the entire application.

The introduction of application platforms, especially in the field of mobile devices, makes the modeling and description of collaborative relationships between individual apps very important. Apps are usually small, isolated working applications that serve a narrowed specific purpose. Multiple apps form a complete system, whereby each individual application serves one specific concern of the overall system. In this context the modeling of collaborations makes synergies visible. The main problem, however, is that application developers can neither know, which other apps are deployed on a target device nor what interfaces they may offer. In addition, no statement can be made about, which applications will be developed and published in the future, which could be used to extend the overall functionality. Unlike the composition of services in a Service-Oriented Architecture (SOA) using orchestration or choreography [20],

 ISBN: 978-1-61208-219-6

mobile platforms do not offer an additional layer to describe composite processes. More, even if such a mechanism would exist, the definition of such composite processes can always only refer to those apps, which are known at design time of the process. That is why the applications themselves need to be able to establish dynamic collaboration relationships.

To realize dynamic, context-aware adaptation, applications need to be modified at runtime. *Role-Oriented Programming (ROP)* [15][26] is an appropriate basis to realize dynamic variation in a fine-grained and collaboration-based manner: This approach of modeling and programming systems is an extension to the classic object-oriented paradigm. The term *role* (or object role) is not related and cannot be compared to the classical role term in workflow systems - also called Role-Based Access Control model (RBAC). In a role universe, core types (i.e., classes) and a set of role types exist. A core object (i.e., instance) can play role instances, if the respective core type and role type are linked by a *can-play-a* relationship. For example, a *Person* can play a *Father* and a *Customer* role type. Players are able to start and stop playing roles at runtime, without losing their own identity. Notably, roles change the behavior of core objects (like aspects in Aspect-Oriented Programming (AOP)) and are able to store additional data, which is only applicable for the respective role and not for the core itself. Beyond that, the role-oriented approach provides a rich repertoire of modeling concepts, which are suitable to describe the semantics of individual exchangeable components and their relationship with respect to the overall system. With ROP roles can be used to model dynamic variation to manipulate the structure, the functionality and the relationships of objects within a single and between several applications.

In this paper, we present a new kind of software architecture—**SMAG** (Smart Application Grid)—in which applications are no longer monolithic, but composed of many small, distributed applications that link to each other dynamically like in a grid. Role-based modeling is used to adapt these applications at runtime, by (a) fine-grained structural and functional changes of application components and (b) dynamically connecting components within one and between several applications. To deal with changes that the application developer did or could not foresee at design-time, exchangeable building blocks are stored in repositories, which can be retrieved and integrated at runtime. This allows the integration of previously unknown components and enables the adaptation to unanticipated scenarios.

This paper is structured as follows: In Section II, we give a short summary about software adaptation. In Section III, we present an overview about the current state of the art w.r.t. dynamic software composition techniques. The Smart Application Grid approach is described in Section IV and an example is described in Section V. Finally, Section VI presents our conclusion and future work.

II. Context-Aware Software Adaptation

Applications that operate in highly dynamic environments, in which context changes and resulting modified requirements often cannot be predicted, can hardly be developed with traditional software development processes. Dynamic adaptive systems (DAS) address this problem by explicit specifications of possible context-triggered runtime changes of the application's functionality and structure during the software development process [6]. The *application context* is defined as the sum of all measurable properties of the application itself and its environment. One possibility is to manually change the application structure, which usually implies high efforts, because of the potential high frequency of context changes and the complexity of the required modification processes. Thus, application architectures are required, that provide runtime support automatically detecting situations, where the application does not match the current context, and to determine and execute the required change-operations to adjust the application accordingly (*adaptive* applications) [24]. Nevertheless, in recent years, numerous development methods, software architectures and adaptation systems for DAS have been developed that base on different approaches [10][12][13][14][17][19], whereof most base on the **feedback loop** [11]. Following this concept, DAS need to (a) identify changed external conditions, (b) analyze the application and its context, (c) make decisions based on those findings on how to adapt and (d) manipulate the system that it fits to the current application context. Another distinguishing factor of DAS is the degree of anticipation w.r.t. the adaptation process. Adaptive systems with *anticipated adaptation* define all possible contextual situations, application variants and their relationship at design time, whereby the system variation is performed at runtime according to predefined rules. In addition, an adaptation to situations that have not been considered during the development process is impossible. This contrasts with applications that support *unanticipated adaptation*. Here, the context model, the analysis mechanisms and the adaptation planning must be open and extensible at runtime. Furthermore, new components—that were not considered at design time—need to be integrated into an application dynamically.

As stated in [13], adaptation can be classified as *parameterized* and *compositional adaptation*. Adaptation by parameterization is achieved by manipulating predefined parameters of software entities. Adaptation by composition refers to the replacement or extension of software components (cf. Sect. III). The adjustment of software elements' parameters is the most trivial solution to adapt an application to a changed environment, which makes the sphere of influence rather limited. It is not possible to actually modify the

 ISBN: 978-1-61208-219-6

functional parts of software units at runtime. Nevertheless, it is possible to change their behavior in a predefined manner by setting prescribed parameters. The implementation code then has to be aware of these parameters, which leads to the problem that adaptation logic is scattered over the application logic. The replacement of software component implementations, on the other hand, makes it possible to actually replace functional and structural elements of an application. When only a small functional part of a system component needs to be altered to adapt it to the application context—for example the enlargement of a message buffer of a communication component—the complete replacement of the component might not be suitable. For that reason, a combination of both adaptation mechanisms is desirable.

To actually adapt an application to a changed environment, two things are essential: the application structure has to be (1) queried and analyzed and (2) the application needs to be modifiable. Under this assumption, component-based software development (CBSD) seems to be best suited for the implementation of adaptive systems. A component-based application consists of reusable components, explicit interfaces and connections. All information that is necessary to use a component is defined in its interface description which will not change frequently. Thus, components having the same interface description are syntactically substitutable, which allows changing the application's functionality by replacing individual components. According to Szyperski "*a software component can be deployed independently and is subject to third-party composition*" [28]. This definition provides the basis to transfer the idea of components-off-the-shelve to DAS. Components are stored in a central repository and grouped by their interface descriptions, so that an application developer can choose from a set of existing components to compose a software system. At runtime those building blocks can be exchanged, based on the current application context as well as the functional and non-functional properties of the different component-implementations.

In this paper, we focus on the process of runtime application modification (i.e., the **act phase** (d) of the feedback loop). We will outline why traditional component replacements lead to problems, how extended software development techniques, like AOP, are able to address them and show the limitations of these extended techniques. Afterwards, we present a novel dynamic software architecture, based on role-oriented modeling, which overcomes these limitations. We will present a role-based composition system, which provides basic functionality for dynamic, unanticipated adaptation.

III. Related Work

This section describes possible methods and state-of-the-art approaches for runtime changes of the structure and functionality of software systems. Our goal is to outline the problems and limitations of known techniques w.r.t. dynamic adaptation.

A. Classical Component Replacement

Some adaptive systems, such as the three-layer energy auto-tuning runtime environment (THEATRE) of the energy auto-tuning approach (EAT) [14], realize (non-)functional runtime variability by exchanging component implementations. This method is based on architectural modeling, where each component type can have multiple implementations. At runtime, for each component type a concrete implementation is chosen to compose the actual system. If the application context changes and the currently selected component is not best suited for it, the adaptation system will replace the specific instance by another component that implements the same component type.

This, however, leads to several problems. Since software components can become very large units, the replacement of a whole component implementation produces much overhead, especially when only a small set of functionality is subject to change. Beyond that, the state of the replaced component needs to be transferred to its substitute or gets lost [12]. Furthermore, all connections between other components and the replaced one need to be updated accordingly. Depending on how long it takes to setup the new component instance, the system state can be temporarily inconsistent.

B. Aspect-oriented Programming

AOP is a paradigm for object-oriented programming, which enables the separation of code fragments that do not relate to the actual core functionality and occur at several points in the program (e.g., security, logging, transaction) from the actual application logic (separation of concerns) [18]. Run-time weavers allow for dynamic aspect integration at execution time. By this means, the application logic can be changed at execution time, making AOP an excellent foundation for implementing dynamic adaptive systems, such as in the DiVA-System [2]. Traditional aspect implementations (e.g., AspectJ [1]) are implemented as white-box code-composition systems imposing a huge complexity for large systems. To cope with this complexity, Model-Driven Software Engineering (MDSE) is used, because model-based representations of an application at runtime allow for an appropriate degree of abstraction [19]. In the DiVA-Project a combination of model-based techniques and AOP was used to support dynamic application variation through aspects, which is called *SmartAdapters* [19]. The SmartAdapters-approach is a generic composition mechanism, but relies on aspects, which still suffer from some fundamental problems: First of all, aspects in general do not have a state. Let us suppose an aspect, which is implementing a message buffer. This aspect would potentially be deployed if the network bandwidth of a communication component decreases. The buffer size, however, would be

 ISBN: 978-1-61208-219-6

implicitly implemented in the advice code and could neither be queried nor changed after the advice was integrated. Second, aspects do not offer a notation to describe collaboration relationships. If the implementation of an advice needs access to the functionality of another component, this connection would be hidden in the advice code. This makes the explicit change of collaboration partners nearly impossible. Because aspect weaving is a code-composition technique, all dynamic variations are class-based modifications. This hinders an instance-based adaptation, because in programming languages like Java all objects share the same method declaration.

C. Context-oriented Programming

Context-oriented programming (COP) is a programming approach based on object-oriented programming that treats context-awareness as a first-class citizen on the level of a programming language [16]. COP extends the collaboration-based dispatch presented in [25] by another context-dimension. Therefore, within a class definition several layers can be declared. Each layer can contain several methods that override given methods of its enclosing class. At runtime the caller of such a method can either explicitly or implicitly specify, whether a layer is active or not. When a layer is active, the call is dispatched to the method of this layer first. By means of this approach, dynamic adaptation through context-dependent dispatch can be realized. Nevertheless, the adaptation-capabilities are rather limited. First of all, there is neither a mechanism to describe dependencies between several layers, nor how layers relate to the application context. The layer activation/deactivation is based on the code that is calling a method on a layer-enabled object. This enables instance-based adaptation because the layer activation/deactivation depends on the context, in which the method of an individual object is called. Like aspects, layers do neither provide a state nor collaboration models. Furthermore, unanticipated adaptation is not possible because layer declarations have to be specified inside a class definition at design-time.

D. Composition Filter

The composition filters approach [7] enables dynamic adaptation by intercepting messages. In contrast to AOP, the composition filters approach achieves the addition and removal of aspect logic, without changing the core class. Furthermore, the conventional object model is extended so that incoming and outgoing messages can be manipulated.

A composition filter class consists of one or more internal classes. Around the composition filter object, an interface part is introduced, which forms the access layer to attributes and methods of the actual core objects. Within this access layer, filters can be added and removed. For each filter, rules are defined that specify, which messages are actually processed and, which are ignored. When a new message arrives, each filter checks if the message is relevant to them. If this is the case, the filter can manipulate the message, forward it to other objects, throw an error or run external code. Subsequently, the potentially modified message will be forwarded to the next filter, until it finally arrives at the actual addressee.

Filters can be dynamically composed in a black-box style and are declared and implemented transparently without the necessity of class-code manipulation. That is why they are very suitable for the implementation of a dynamic composition system and realization of dynamic adaptive applications. In Section IV, this approach is extended with explicit collaborative relationships, by combining composition filters with role-based programming.

IV. Smart Application Grids

Smart Application Grids consist of many small, distributed applications that are linked dynamically. To adapt these composite applications at runtime to context changes, the *individual* applications are dynamically modified and *collaboration relationships* are changed depending on external conditions (i.e., conditions of the context) using roles. The main goal is to develop a composition system that is transparent, supports reuse of composition programs and is technology-independent. In addition, through metamodeling it should be possible to investigate the application structure using models at runtime, where model changes should affect the running application transparently. This paper specifically focuses on the underlying dynamic role-based composition system, not on specific adaptation mechanisms. Therefore, first the concepts of role-based modeling are summarized and, subsequently, the concepts of our proposed role-based composition system as well as the associated repository are introduced. According to Aßmann, a composition system consists of a component model, a composition technique and a composition language [5]. In the following, the SMAG composition system is presented according to these three aspects. Finally, an exemplary adaptation architecture and our reference implementation in Java are presented.

A. Role Modeling

Although there is no uniform understanding of the concept of roles in software engineering, a consensus has emerged that roles are an important element of software design. In this work the concept of a *role* is used according to the work of Riehle [23] and Reenskaug [21] and is briefly summarized in this section.

A **role** is a *dynamic view* or a *dynamic service* of an object in a *specified context*, offering the possibility of separation of concerns, interface-structuring, dynamic collaboration description, and access restriction. A role is clearly specified by a **role type** and can be played and removed from an object at runtime. When an object plays a role, they both share the same identity (i.e., a role does not

 ISBN: 978-1-61208-219-6

have its own identity), whereas the number of roles played simultaneously is not limited. It is also possible that a role plays other roles. Let us consider a role `Employee` which is played by an instance of a class `Person`. This role could play another role of the type `Software Developer`. Roles have their own properties and methods, whereas the role-playing object behaves according to the functionality, defined by the role. Furthermore, the object state is extended by the properties of the roles it is playing. If an object of the class `Person` is playing the `Employee` role, it might get an additional attribute `salary` and an additional method `work()`. Any call to the core object is first dispatched to its roles. Because roles can have references to other roles and because they must be played by objects, roles provide means for describing dynamic collaborations, spanning a varying network of dynamic relationships.

In a class diagram, classes and their relations are modeled. Analogously, a **role model** specifies role types and their relationships. In this way, it describes object collaborations, since the instantiated roles necessarily have to be played by objects or other roles. Usually classes expose public methods, which are used to establish interaction between objects (i.e., instances of those classes) by exchanging messages (i.e., calling methods and passing parameters). Associations in class diagrams however, neither provide means for describing under which circumstances objects collaborate nor which part of its interface (i.e., set of attributes and methods) is relevant w.r.t. this relationship. A role model on the other hand, describes only a single concern of the object collaboration which allows for separation of concerns. Role models can then be composed hierarchically to express multi-concern object collaboration [22]. This increases the degree of reuse because domain dependencies can be controlled in a fine-grained manner. In this way, partial architectures can be specified, shared and reused.

Subsequently, role and class models are merged to **role-type-class-models**, by binding all role types to classes. Riehle describes several role restrictions that can be used to control, how role types can be applied to classes [23]. The set of all role types of a class is called role-type set which specifies what types of roles an instance can play. Traditionally in class modeling, static associations are used to express possible interaction relationships between objects. The modeling of object collaborations through role types allows for a fine-grained description of dynamic collaborations between several objects under certain conditions (role context). Through this modeling approach, both interactive relations that change at runtime and the subset of methods that are involved in this interaction can be documented. Figure 1 gives an example. The two role models ”*Strategy*” and ”*Observer*” are composed to new role model ”*Strategy and Observer*”, by introducing a role-equivalent restriction. This restriction claims that every object that plays a role of the role type *Strategy* must also be capable to play a role

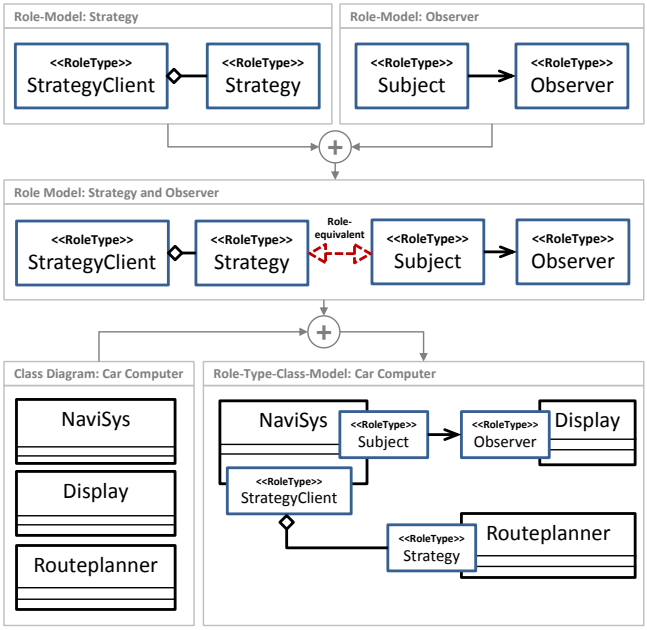


Figure 1. The Relationship between the SMAG Component Model and Role-Based Modeling.

of the role type *Subject* and vice versa. This role model is then combined with a class diagram ”*Car Computer*” to a role-type-class-model. This model describes, which class instances are able to play a given set of roles. Lets consider an instance of the class *NaviSys* is playing a role of the role type *Subject*, it is related an object playing the role *Observer*. After the role has been removed, this relationship is removed either.

The role-based software development approach is primarily an approach on the modeling level, whereas different approaches were suggested on how to implement the role-based approach. The lack of a silver-bullet solution leads to a deep gap between design and implementation. There is the possibility of using classical object-oriented concepts, such as interfaces, multiple inheritance or mixin inheritance to realize the presented concepts [27]. However, this means that roles can not be acquired or removed at runtime. Another solution is the Role Object Pattern (ROP) [8] which separates the core and the role objects into different classes, using delegation to interact between them. This leads to a nontransparent role-binding mechanism, because the core object has to manage its roles. In addition, several attempts have been made to integrate the role concept in programming languages. ObjectTeams/Java (OT/J) [4], an extension of the Java programming language and part of the Eclipse platform, is one of them. Despite the current OT/J implementation does not fully realize the role-concept described earlier, it was still used for the reference implementation of SMAGs for Java (cf. Section IV-E), due to the lack of mature alternatives.

 ISBN: 978-1-61208-219-6

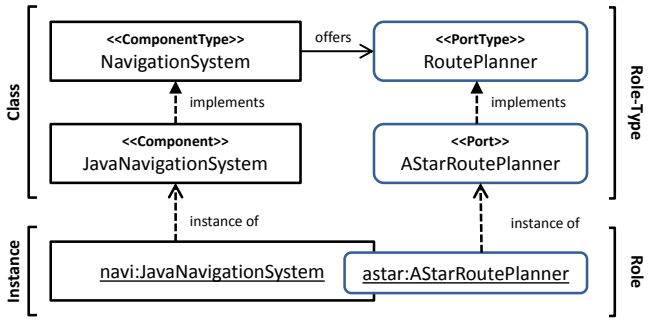


Figure 2. The Relationship between the SMAG Component Model and Role-Based Modeling.

B. Component Model

SMAG Components are stateful, self contained software modules that can be developed and deployed independently. They are described by a **ComponentType** which describes the functional interface of a component by grouping several **PortTypes**. A PortType represents a language-independent interface description which can be offered or required by a component. Each PortType has a unique name and specifies the services of those components that provide this interface. It defines both a functional view by method signatures and a state-based view by a list of externally manipulable parameters. Each PortType is implemented by one or several **Ports** which are the elements that can be added or removed at runtime. There are, however, certain state attributes that have to be preserved in spite of the substitution process. To realize this requirement, without the use of elaborate mechanisms for transferring each individual state value, a PortType can specify a set of attributes (*SharedMemory*) that are managed within a component and not within a replaceable unit. Furthermore, a PortType can specify a number of other PortTypes that are required to provide the desired functionality. Each required PortType is annotated with a multiplicity to specify how many instances can be managed by a Port. In analogy to the ACOE-component model [10], PortTypes are further distinguished in *BehavioralPortTypes* and *EventPortTypes*. A BehavioralPortType allows access to application functionality defined by it, whereas an EventPortType provides an event at which other components can register on. Besides the actual application functionality, a Port which is implementing a given PortType, must declare some metadata to give information about whether it is suitable for the requirements of the specific domain. As mentioned earlier, the composition system is optimized for a possible runtime adaptation. Therefore, metadata should be automatically computable. This can be achieved by using semantic technologies like URIs to concepts of shared ontologies [9]. Finally, a component is implemented, using a specific programming language. First, the language-independent ComponentType declaration is transferred into a language artifact. This is done automatically using a platform-specific IDL compiler. The component can offer a standard-implementation for any method that was specified in the provided Port-Types. In addition, they can implement an install script and an uninstall script that is executed when the component is instantiated or destroyed. In this way, any required resources can be allocated or released.

As already mentioned, the presented composition system implements the ideas of role-based software development with the semantics of the composition filter approach [7]. Figure 2 shows the relationships of the elements of the component model to the concepts of role-based designs. A PortType corresponds to a Role-Type, a Port to a Role, a ComponentType to a Class and a Component to an object. Under this assumption, the specification of PortTypes coincides with the creation of a role model, whereas the required PortTypes of a given PortType form the collaborations. The declaration of ComponentTypes is similar to the design of a class-role-type-model, as PortTypes, representing role types, are mapped to ComponentTypes. At runtime, an instance (Component) may play roles (Ports), thereby changing its behavior.

The proposed component model defines software components as modular units with explicitly defined interfaces. Thereby, components that implement the same ComponentType as well as Ports that implement the same PortType are syntactically substitutable. Initially when a component calls a method of a connected component via a required/provided PortType, the base implementation of this method within the actual component is executed. When however, a Port is bound to a provided PortType of a component, then the method implementation of this Port is called first. As described in the next section, the Port can process the method call completely or partially, by adding additional functionality and passing the call to the underlying implementation (i.e., another Port or the base implementation of the Component). Ports can be bound to Components dynamically, thus changing their structure and behavior w.r.t. to a specific PortType. Ports in addition, offer internal state variables as well as external parameters, to tailor them w.r.t. to a concrete reuse context.

C. Composition Technique

At runtime, the required PortTypes of a component instance have to be connected with offered PortTypes of other components. This is done by **passive connectors**. In some systems, architectural connectors are active elements which perform type conversions or protocol adjustments. As we will explain, these requirements can still be implemented, using specific Ports. Components can either be extended by inheritance at design-time or by an *extend operator* at runtime. The extend operator introduces new PortTypes, by manipulating the architectural model that is managed

 ISBN: 978-1-61208-219-6

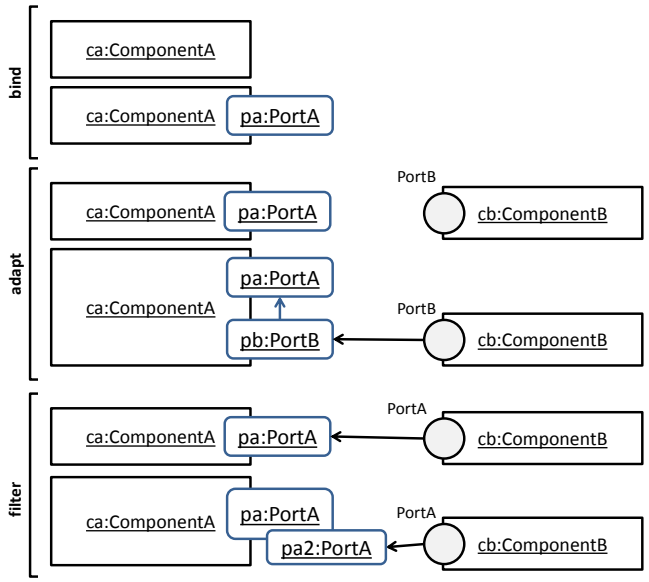


Figure 3. The Bind, Adapt and Filter Composition Operators with an Example.

dynamically. Each change in the architectural model is then transferred to a change of the corresponding application.

The main composition operators of the presented composition system are the *bind* and the *unbind operator*, which bind a Port to a Component respectively removing a Port from a Component. By binding a Port to a Component, all invocations of methods that are specified in its PortType will first be dispatched to the Port. Depending on the implementation, the Port can provide the whole functionality or just performs some extra tasks and then forwards the call to the actual component or an underlying stacked Port. By removing a Port, the internal state gets lost. For this reason, each component manages a shared memory which keeps state information regardless of the existence of bound Ports. In the presented approach, connectors can connect provided and required Ports only, if they share the same PortType. In practice, however, it is common that the interfaces of reused components differ from the needed ones. In this case, an interface adaptation can be carried out by special *AdapterPorts*. An AdapterPort is usually a lightweight Port that implements exactly one PortType and has exactly one required PortType. Figure 3 illustrates this concept. In the implementation of the adapter, the two interfaces can be mapped by translating each call to the offered interface Pa to an appropriate call to the required interface Pb. A special case of adaptation is the remote communication between distributed components. If the required functionality is provided by a component that is not deployed on the same system, an AdapterPort can automatically be generated which translates each local to a remote call.

To separate cross cutting concerns (e.g., persistence, logging, security) from the actual application functionality, which is realized through Ports or the component implementation, *FilterPorts* can be used. A Filter implements a given PortType and requires a Port of the same type. In analogy to the role modeling, in which a role can also play other roles, a port instance can be decorated with several FilterPorts. Depending on the implementation of the filter, both incoming and outgoing messages can be processed. Figure 3 shows this process schematically. Furthermore, FilterPorts can be used to offer a set of methods of a component to other applications by providing a remote interface. Therefore a RemoteFilter would publish some kind of remote interface (e.g., as a web-service) during the binding process. This service can then be published in a central repository, making the offered functionality visible to other applications. This creates a dynamic heterogeneous network of collaborating applications.

D. Composition Language

For developers of SMAG applications, two different modeling levels are available, on which they can describe architectures. On the one hand, there is the possibility to specify metaarchitectures, which are modeled by only using Component- and PortTypes. As explained earlier, this approach is in direct connection with the creation of a role-model and the generation of role-type-class-model. Nevertheless, no statement is made, which concrete Component implementations will be used and what Ports will be deployed at runtime. Since Component- and PortTypes are independent of a concrete programming language, which are implemented by artifacts of a concrete platform, it is possible to specify platform-independent and reusable architectures. Those partial architectures can be reused through model composition.

On the basis of the metamodel, an architectural model can be specified by choosing a component implementation for each ComponentType. At runtime, the architecture description is preserved, so the runtime environment can create a direct connection between the instantiated software artifacts and corresponding model elements. The runtime environment is always able to translate changes in the application model to changes in the application itself. In general, object-oriented programming languages support introspection, enabling the runtime environment to collect information about the objects. In most cases this approach can also be used to derive an application model at runtime which leads to the realization of a complete round-trip. When both strategies are applied, it can be ensured that the architectural model and the structure of the actual application are synchronous.

E. SMAG-Repositories

To reuse SMAG model elements and software artifacts efficiently and finding them at runtime, they must be published in a central location. This idea is not new and

 ISBN: 978-1-61208-219-6

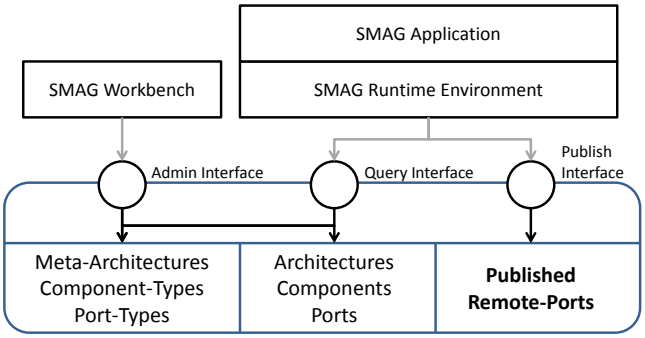


Figure 4. SMAG-Repository Structure.

has already been implemented several times. Thus, web services for example can be published and discovered using a uniform standardized directory service, called Universal Description, Discovery and Integration (UDDI). Although UDDI has been standardized by OASIS, it did not become very popular. One of the main reasons is the required effort implied by large specification and the complex process of service publication. For this reason, we tried to keep the specification of SMAG-Repositories as simple as possible.

A SMAG-Repository has three main tasks. On the one hand, artifacts must be published and managed as well as searched and reused. On the other hand, applications must be able to publish functionality at runtime that can be used by other SMAG applications. SMAG repositories consist of a static directory for Component- and PortTypes, MetaArchitectures, Architectures, Components and Ports. Each artifact is identified by a unique URI and can be described using various metadata that is made computable by referencing OWL concepts. Through an administration interface they can be published, unpublished or modified. Components and PortTypes, MetaArchitectures are stored as platform-independent models, whereas Components and Ports are stored as platform-specific binary packages. A publish interface allows a SMAG application to provide a remote service by specifying a PortType, a calling address and a description of the used communications technology. Other applications are able to query this directory services, which enables the setup of dynamic collaboration between various applications. This corresponds to the implementation of a Trader Service, which is looking for a service based on its functional properties and characteristics. Each repository can reference an unlimited number of partner directories, so queries can be forwarded if a repository is not able to deliver results for a given request. Figure 4 illustrates the structure and relationship of the SMAG repositories. Each SMAG Repository in turn is a SMAG application whose architecture is based on a fixed metaarchitecture. Thus, it can be implemented for different target systems and dynamically changed, according to specific needs. The three public interfaces (a) administrative, (b) search and (c) runtime publication, are made available through RemotePorts.

F. Reference Implementation

The idea of Smart Application Grids is basically platform- and technology-independent. The model-based description of ComponentTypes and Ports does not make any statements about a specific programming language. However, Components and Ports will be implemented in a specific language using a specific runtime environment. Thus, the component model has to be mapped to concepts of the target language, whereas the runtime environment has to implement the given composition operators.

A reference implementation was created using the Java programming language in combination with the role-based language extension ObjectTeams/Java. Accordingly, ComponentTypes and PortTypes are mapped to Java interfaces, Components to classes and role models to abstract OT/J-Teams. Ports are implemented through specific OT/J-Roles. Since OT/J currently only supports load-time-weaving, theoretically, no new roles can be added after the class, that should play the role, was loaded. This hinders the promised support for the dynamic introduction of previously unknown Ports. To circumvent this drawback, a proxy team is generated, which contains proxy roles for each PortType. The Ports are added to the proxy teams at runtime by delegation. This workaround, however, does satisfy all presented requirements regarding transparent role playing. To perform the composition operations at the application level, the runtime environment must manage a model representation of the application architecture. For both the metaarchitecture and the architecture, Ecore-based metamodels were created. Based on this representation, models are used to query and modify the application at runtime. By model-to-text transformation, all Java artifacts can be generated, that are not reused and all reused elements can be obtained from a repository.

V. Example

To demonstrate the presented results, an exemplary adaptation system was developed. Figure 5 shows the basic architecture. First, a distinction is made in a real and a virtual context, whereas sensors (sensor layer) monitor the real-world context and transfer measured values into an object-oriented representation. The inference layer is notified of context changes. It then can put the different context characteristics into relation or generate new knowledge. The adaptation layer makes sure that the running SMAG application fits the current application context. Therefore, the application architecture is queried using the query interface of the SMAG runtime environment and is compared to the requirements that result from the context. Subsequently, if necessary, it creates a reconfiguration script that consists of individual reconfiguration operators that are executed against the manipulation interface of the runtime environment. The

 ISBN: 978-1-61208-219-6

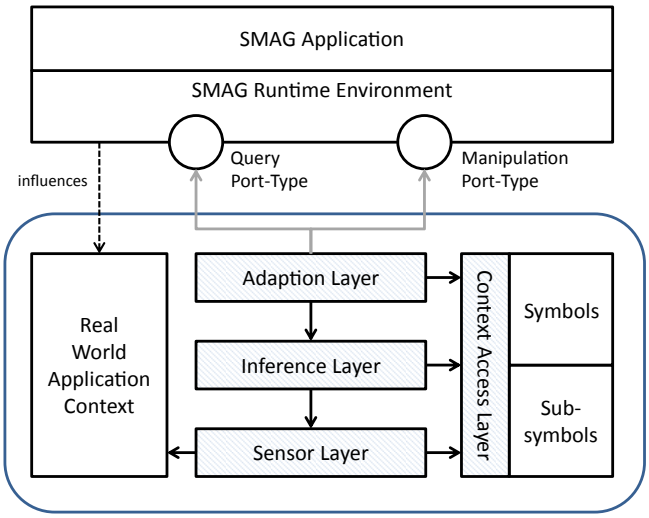


Figure 5. The General Architecture of the SMAGs-Adaption System.

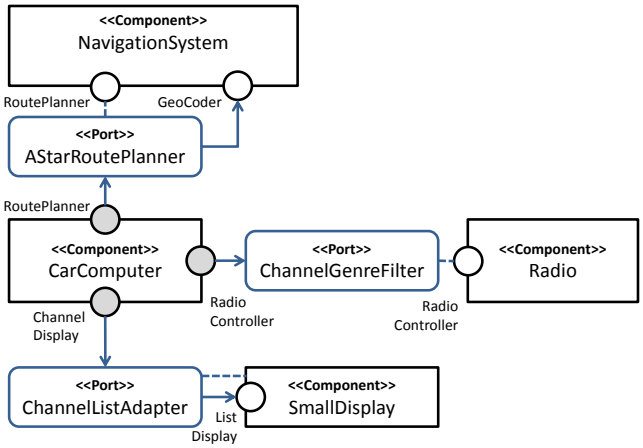


Figure 6. A Simplified Diagram of the Run-Time Architecture of an Example Application.

adaptation system was implemented as a SMAG application. This allows to alter the algorithms and mechanisms that are used to gather and process information as well as the execution of adaptation operations at runtime. This, however, in combination with the possibility to integrate Ports that were not known at design-time, enables the realization of unanticipated adaptation. The adaptation methods that were used for evaluation, are based on ECA-rules using the JBoss Drools system [3]. Nevertheless, this approach is neither new nor sophisticated, it is still suited to ensure the proper functioning of the dynamic composition system.

In Figure 6, the runtime architecture of an example system is shown, which consists of four components: `CarComputer`, `NavigationSystem`, `Radio` and `SmallDisplay`. The `NavigationSystem` component offers a PortType `RoutePlanner`, which provides the functionality to calculate a route. In the shown configuration, a port is attached to the component, which is implementing an A*-algorithm, based on locally stored maps. When the memory of the system exceeds a specified threshold and an Internet connection is available, this port is dynamically replaced by an implementation, which is using a web service.

The Radio component provides an interface to query all radio channels that are available. Depending on, which driver was detected an whether any profile information is available, a channel filter can be deployed dynamically, which only shows radio channels relevant to the current driver. When the driver changes, the `RadioGenreFilter` is either parameterized with another genre or is removed.

The `SmallDisplay` component however provides an interface to display a list of strings, whereby the `CarComputer` component requires an interface to display a list of radio channels. Those interfaces can me mapped using an adapter port.

VI. Conclusion and Future Work

In this paper, it was shown that role-based modeling can be transferred to component-based software design, in order to create applications that can be manipulated in a very fine-grained manner at runtime. Furthermore role-based modeling is putting emphasize on dynamic collaborations, which makes it possible to create dynamic relationships between several components within one application and between several ones. The proposed SMAGs concept allows to create clearly structured and reusable application architectures using role-based modeling, which makes it an ideal candidate for the realization of dynamic adaptive systems. This is because roles are stateful, functional units with clearly defined interfaces that can be dynamically merged with objects. Notably, the role-based modeling approach can be connected with the basic principle of composition filters very well. The implementation effort of SMAG applications is limited to the implementation of the actual application logic, since the majority of software artifacts can be generated automatically using models. Even though the presented adaptation mechanisms are merely exemplary, the adaptation logic could be clearly separated from business logic and the software system could be changed at runtime, supporting unanticipated dynamic adaptation. The concept is largely based on the use of a central repository to manage all modeling and implementation artifacts. In this way, implementations of ports that were not known during design-time can be integrated into a running application or old versions of existing ports can be replaced with newer ones, enabling hot updates.

However, many open issues remain that need to be refined in future work. The most important aspect is the support of the development process by appropriate tools. Furthermore, it needs to be discussed, how the consistency of an application, in terms of structural changes by composition operations,

 ISBN: 978-1-61208-219-6

can be guaranteed. Since the concept describes structural changes at the level of architectural models, model-based validation would be appropriate. In addition, the semantic substitutability of different port implementations needs to be ensured. This corresponds to the fundamental problem of trust within components-of-the-shelf, so it needs to be evaluated, whether certification can solve this problem. The presented adaptation architecture needs to be equipped with advanced adaptation mechanisms, based on the semantic description of context values. In addition, it may be possible to correlate the role context, in which an object is playing a role, with the application context, to automatically deploy role bundles and maybe enable autonomous and unanticipated dynamic adaptation.

ACKNOWLEDGEMENT

This research has been funded by the European Social Fund and the State of Saxony (Project ZESSY #080951806).

REFERENCES

[1] AspectJ. http://www.eclipse.org/aspectj/. 11.05.2012.

[2] DiVA-Project. http://www.ict-diva.eu/. 11.05.2012.

[3] JBoss Drools. http://www.jboss.org/drools/. 11.05.2012.

[4] OT/J. http://www.eclipse.org/objectteams/. 11.05.2012.

[5] U. Aßmann. *Invasive Software Composition*. Springer Verlag, New York, USA, 1st edition, 2003.

[6] N. Bencomo, P. Sawyer, G. S. Blair, and P. Grace. Dynamically adaptive systems are product lines too: Using model-driven techniques to capture dynamic variability of adaptive systems. In *SPLC (2)*, pages 23–32, 2008.

[7] L. Bergmans. The Composition Filters Object Model. Technical report, Dept. of Computer Science, University of Twente, 1994.

[8] D. Bumer, D. Riehle, W. Siberski, M. Wulf, and M. Wulf. The role object pattern. In *Washington University Dept. of Computer Science*, 1997.

[9] M. Cremene, M. Riveill, and C. Martel. Unanticipated dynamic adaptation of context-aware services. *Acta Tehnica Napocensis*, pages 30–35, 2008.

[10] B. Ding, H. Wang, D. Shi, and X. Rao. Towards unanticipated adaptation: An architecture-based approach. In *Software Engineering Research, Management and Applications, 2009. SERA '09. 7th ACIS International Conference on*, pages 103 –109, 2009.

[11] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.

[12] P. Ebraert, T. D'Hondt, Y. Vandewoude, and Y. Berbers. Pitfalls in unanticipated dynamic software evolution. In *RAM-SE*, pages 41–50, 2005.

[13] J. Fox and S. Clarke. Exploring approaches to dynamic adaptation. In *Proceedings of the 3rd International DiscCoTec Workshop on Middleware-Application Interaction*, MAI '09, pages 19–24, New York, NY, USA, 2009. ACM.

[14] S. Götz, C. Wilke, S. Cech, and U. Aßmann. Runtime variability management for energy-efficient software by contract negotiation. In *Proceedings of the 6th International Workshop Models@run.time (MRT 2011)*, 2011.

[15] G. Guizzardi. *Ontological foundations for structural conceptual models*. PhD thesis, Enschede, 2005.

[16] R. Hirschfeld, P. Costanza, and O. Nierstrasz. Context-oriented programming. *Journal of Object Technology, March-April 2008, ETH Zurich*, 7(3):125–151, 2008.

[17] M. U. Khan. *Unanticipated Dynamic Adaptation of Mobile Applications*. PhD thesis, University of Kassel, March 2010.

[18] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In *ECOOP*, Berlin/Heidelberg, 1997. Springer Verlag.

[19] B. Morin, O. Barais, G. Nain, and J.-M. Jezequel. Taming dynamically adaptive systems using models and aspects. In *Proceedings of the 31st International Conference on Software Engineering*, ICSE '09, pages 122–132, Washington, DC, USA, 2009. IEEE Computer Society.

[20] C. Peltz. Web services orchestration and choreography. *Computer*, 36(10):46–52, Oct. 2003.

[21] T. Reenskaug, P. Wold, and O. A. Lehne. *Working with objects - the OOram software engineering method.* Manning, 1996.

[22] D. Riehle. Describing and Composing Patterns Using Role Diagrams. In *Proceedings of the 1996 Ubilab Conference*, Zurich, 1996.

[23] D. Riehle and T. Gross. Role model based framework design and integration. In *Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, OOPSLA '98, pages 117–133, New York, NY, USA, 1998. ACM.

[24] R. O. Rossen and R. Rashev. Adaptability and Adaptivity in Learning Systems, 1997.

[25] Y. Smaragdakis and D. Batory. Mixin layers: an object-oriented implementation technique for refinements and collaboration-based designs. *ACM Trans. Softw. Eng. Methodol.*, 11:215–255, April 2002.

[26] F. Steimann. On the representation of roles in object-oriented and conceptual modelling. *Data Knowl. Eng.*, 35(1):83–106, 2000.

[27] F. Steimann. Role = Interface: a merger of concepts. *Journal of Object-Oriented Programming*, pages 23–32, 2001.

[28] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

 ISBN: 978-1-61208-219-6

Adaptive Properties and Memory of a System of Interactive Agents: A Game Theoretic Approach

Roman Gorbunov, Emilia Barakova, Rene Ahn, Matthias Rauterberg
Designed Intelligence Group, Department of Industrial Design
Eindhoven University of Technology
Eindhoven, Netherlands
r.d.gorbunov@gmail.com

***Abstract*—In this work we present an evolving system of agents which interact with each other in game theoretic settings. The parameters of the game are considered as time dependent variables representing state of the external environment. The variation of these parameters covers four canonical examples from game theory: prisoners dilemma, hawk-dove, stag-hunt and harmony game. The process of self-adaptation of the proposed system as well as its adaptation to the changing parameters of the environment is considered. We have demonstrated that the introduced system can be in different states which determine the way the system adapts to the external conditions. Stability of these states with respect to the variation of the external conditions has been studied. An ability of the system to accumulate memory about its past experience has been reported.**

***Keywords*-complex adaptive systems; game theory; evolutionary game theory; Turing machines**

I. INTRODUCTION

A complex adaptive system (CAS) is a collection of autonomous, heterogeneous agents, whose behavior is defined by a limited number of rules [1]. In this work we consider a special class of CAS in which agents have a fixed and limited number of actions available to them and outcomes of every pairwise interaction between agents depend on the actions chosen by the two agents participating in the interaction. In more detail, in every pairwise interaction every agent receives a payoff which depends on the actions chosen by the considered agent as well as on the action chosen by another agent involved in the interaction. The mathematical constructs of this kind are typically studied within game theory. However, when the number of agents becomes large, the system starts to exhibit complex and interesting emergent properties which are hard to derive from the underlying simple game theoretic rules of the interactions between the agents and, as a consequence, methods of CAS can be better suited for study of collective properties of the system.

The complexity of the system is conditioned by large number of heterogeneous agents. The behavior of the system as a whole is hard to derive from individual properties of agents and rules of the underlying game. In this framework the payoffs of the game can be considered as parameters of the environment into which the system of interacting agents is embedded. If we supply such a system with a selection mechanism, such that the agents with the largest fitness survive, the populations of the agents of different types will depend on parameters of the game (environment). This property makes the system adaptive. Moreover, the fitness of a given agent in the system depends not only on the parameters of the game but also on the proportion of agents of different types. This makes the system self-adaptive.

	A_1	A_2
A_1	w, w	x, y
A_2	y, x	z, z

Table I
PAYOFF MATRIX OF A SYMMETRIC GAME

In this work we consider a simple game in which every agent has only two actions available. Moreover, we consider symmetric games in which payoffs of two interacting agents depend on their actions and actions of their opponents in the same way. The payoffs in a symmetric game are given by the payoff matrix shown in Table I. The rows and columns of the table correspond to the two actions available to the first and the second agents, respectively. The first and the second number in the cells are payoffs of the first and second agents, respectively. We do not change the logic of a game if we multiply all the payoffs by the same positive number. The logic of the game is also invariant with respect to a constant shift of all payoffs. We also can change the numeration of the actions. Using these three properties we can always set $w = 0$ and $z = 1$. As a consequence we need only two parameters (x and y) to completely specify the game.

Changing x and y we will get different games. Four qualitatively different games can be identified. If $x \in (1.0, 2.0)$ and $y \in (0.0, 1.0)$ we have a game which is known in biology and evolutionary game theory as hawk-dove game. In political science and economics the games is more known as chicken game. The earliest presentation of a form of the hawk-dove game was by John Maynard Smith and George Price in their 1973 Nature paper, "The logic of animal conflict" [2]. In biology this game formalizes a situation in which there is a competition for a shared resource and

 ISBN: 978-1-61208-219-6

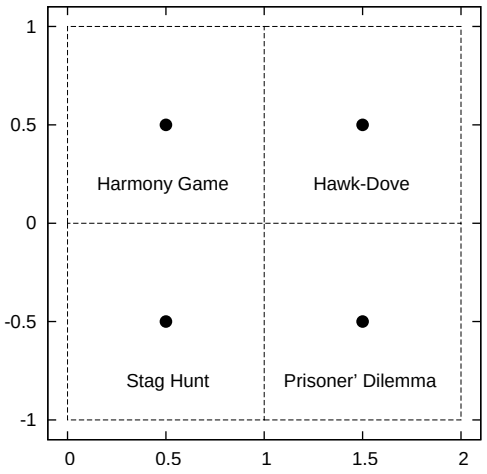


Figure 1. For regions corresponding to four different games

the contestants can choose either conciliation or conflict. Sometimes this game is also referred to as snowdrift game. In game theory this game is known as a canonical model of conflicts between two players and has been the subject of extensive research [3]. The hawk-dove game was also used to compare approaches of CAS and game theory to analysis of systems of interacting agents [1].

If $x \in (1.0, 2.0)$ and $y \in (-1.0, 0.0)$ we get the prisoners dilemma game [4], [5] which is known a canonical tool to study cooperative behavior in game theory, politics, economics, sociology and evolutionary biology. Many natural processes have been abstracted into models in which living beings are engaged in repeated games of prisoner's dilemma. In this game agents need to make a choice between two actions which are called "cooperation" and "defection". The main idea behind the prisoner's dilemma game is that the defection is always more beneficial than the cooperation, independently on what strategy is chosen by the opponent. On the other hand, the mutual cooperation of the agents is more beneficial to both players than the mutual defection. Because of these two properties the game faces players with a dilemma between the cooperation and defection. The iterated prisoner's dilemma has been considered as a part of a model that explains how cooperation can arise in the nature as a result of evolution [6]. The tit-for-tat strategy has been reported as the best deterministic strategy for the iterated prisoner dilemma [7] demonstrating that the prisoner's dilemma can be used to explain reciprocity.

If $x \in (0.0, 1.0)$ and $y \in (-1.0, 0.0)$ we get the stag-hunt game which describes a conflict between safety and social cooperation [8]. This game is also known as "assurance game", "coordination game", and "trust dilemma". Several animal behaviors have been described as stag-hunts including coordination of slime molds and hunting practices of orcas.

If $x \in (0.0, 1.0)$ and $y \in (0.0, 1.0)$ we get the harmony game. This game is trivial since it does not induce any dilemma. It is included into consideration for the completeness of the classification.

In the introduced games the agents always have two choices. These choices can be called as cooperation and defection. By definition the mutual cooperation is always more beneficial than the mutual defection.

II. METHODS

Additionally to the payoff matrix we need to specify a schema of interaction between agents. There are different ways to organize the interaction between agents. For example, in the so called proposer-responder settings, the first agent, called proposers chooses one of the two actions. The second player, called responder, can see the choice of the proposer and based on that it makes its own choice. After that the choices of the two agents are combined to calculate the payoffs that have to be allocated to the two agents. In our model we take an alternative approach which treats agents in a symmetric way. The two agents make their choices simultaneously. After that their choices are reviled to each other and are used to allocate the payoffs to the agents.

Another important component of the interaction schema is the number of games. If any two agent play with each other only once we have the case of a single stage game or single shot game. Alternatively we could have a case of repeated games. The repeated games can be broadly divided into two classes: finitely and infinitely repeated games. In the case of the finitely repeated games the agents interact with each other a fixed and know number of times. In case of the infinitely repeated games the number of interactions is not fixed. There is a non zero probability, usually fixed and constant, that after the given interaction there will be another interaction. The repeated settings of the game have been shown to be one of the possible mechanisms that can stimulate emergence of cooperative behavior as a result of evolution [9], [10]. In our model we use the settings of the infinitely repeated games. The probability for the next game to happen was set to 0.5.

Additionally to the payoff matrix and the interaction schema we need to specify how agents are paired to interact with each other. One of the options is to assign special location to every agent and let the agent to interact with its neighbors. In this case we get the case of spatial games [11]. The effect of space can have a strong effect on the dynamics of the evolution. In particular agents of certain class can be more successful than other agents because of the fact that they can form spatial clusters in which they mostly interact with the agents of the same kind. This kind of effects supplements the selection of agents by selection of groups of agents and can promote cooperative behavior. In our model we do not consider effects of the spatial locations to focus more on dynamics of relative proportion of different

 ISBN: 978-1-61208-219-6

types of agent in the population, which exhibits a complex and interesting behavior worth of separate consideration.

The payoff of an agent depends on what agents were involved into the interactions with the given agent as well as how many games were played with every opponent. In other words the fitness of a given agent is a stochastic property. In the presented model we calculate the average fitness of every agent in the given population of the agents. This assumes that every agent lives long enough to have many rounds of interactions with any other agent. In mathematical terms the fitness of agent i is given by the following equation.

$$f_i = \sum_{j=1}^{n} \nu_j \sum_{k=1}^{\infty} \mu_k \cdot p_{i,j}(k), \qquad (1)$$

where the first summation is over all kinds of agents in the population, ν_j is the portion of agents j in the population. The second sum is over the number of interaction in a single round of interactions between the agents i and j. The μ_k is the probability that there will be k interactions in one round of interactions and $p_{i,j}(k)$ is the payoff of agent i while interacting with agent j in k games.

To start the evolution of the system we need to specify initial relative fractions of agents of different kinds. After that we calculate the average fitness of agents of different kinds using the above given equation (1). Different kinds of agents have different fitness depending on the parameters of the environment (payoff matrix of the game) and portions of the agents of other types. In other words the agents of the same kind can have different fitness in the same game depending on the frequencies of agents of other kind in the population. In this sense the evolutionary process makes the system of agents not only adaptive to the external conditions but also self-adaptive.

On every step of the evolutionary dynamics we calculate average payoff for all kinds of agents in the population. These payoffs determine the changes of the relative proportions of the agents. The proportion of those agents which get a high payoff increases while the proportion of the low-payoff agents decreases. The standard equation giving the changes of the proportions of the agents of different types as a function of their payoffs and proportions of agents of other types is called replicator equation and has the following form:

$$\frac{d\nu_i}{dt} = \nu_i \cdot [f_i(\nu_1, \ldots, \nu_n) - f(\nu_1, \ldots, \nu_n)], \qquad (2)$$

where f is the average payoff in the population:

$$f(\nu_1, \ldots, \nu_n) = \sum_{i=1}^{n} \nu_i \cdot f_i(\nu_1, \ldots, \nu_n) \qquad (3)$$

According to this equation the proportion of agents whose payoff is larger than the average one will increase. The growth of a population of agents of a given type is proportional not only on the relative payoff of these agents but also to the current relative size of their population.

The average payoff of the agents grows with the probability of next game in the repeated games settings since the average number of games increases. As a consequence the system will evolve faster if average number of games is larger. To make the average speed of evolution insensitive to the probability of next game we have used a modified version of the replicator equation:

$$\frac{d\nu_i}{dt} = s\nu_i \cdot \frac{f_i(\nu_1, \ldots, \nu_n)}{\max(f_1, \ldots, f_n) - \min(f_1, \ldots, f_n)} \qquad (4)$$

According to the modified version of the replicator equation the growth of a subpopulation of agents of a given type depends on the position of their average payoff relative to the maximal and minimal payoffs in the whole population. In front of the right part of the equation we have added a parameter s which can be considered as a time step for the numerical simulation of the evolution. It can be set to a value smaller than 1 to make evolution slower and smoother. In our calculations we use $s = 0.4$.

III. Results

A. Evolution with Two Types of Agents

In case of a population consisting of only two types of agents, four qualitatively different kinds of evolutions are possible depending on how payoffs of the agents depend on the portions of the two subpopulations in the whole population. If the average payoff of agent i while interacting with agent j is p_{ij} then payoff of the agent i in the population is equal to:

$$p_i = \nu \cdot p_{i1} + (1 - \nu) \cdot p_{i2}, \qquad (5)$$

where i can be 1 or 2 and ν is the proportion of the first agent in the population. In the Figure 2 we have shown four qualitatively different relations between the payoffs of the agents of the first and second type. In the case of divergence (left top tab) agents of a given type have larger payoffs if their portion is large enough. In this case their portion increases and the agents of another type are completely eliminated from the system. In the case of convergence (right top tab) the agents of a given type have larger payoff than those of the agents of another type if their portion is small enough. As a consequence the portion of those agents which have a larger payoff increases until the convergence points at which both agents get the same payoffs. At this point agents of two types coexist. Another case is absolute domination. This situation happens when one of the two types of agents has larger payoff than another type agents independently on the portion of the agents in the populations. On the left bottom tab we have a case in which agents of the first type absolutely dominate the agents of the second type. In this case the agents of the second type always

 ISBN: 978-1-61208-219-6

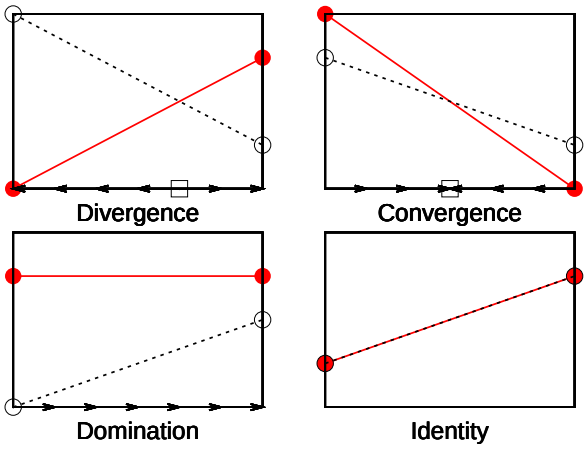


Figure 2. Schematic representation of four possible scenarios of evolution for a system consisting of two kinds of agents. The x-axis is the portion of the agents of the first kind in the population. The red solid and black dashed lines are payoffs of the agents of the first and second kind, respectively.

became extinct. Finally we can have a case when two agents always have the same payoff (see the right bottom tab). In this case the proportions of agents either do not change or exhibit a random walk (depending on the model used for the evolutionary process).

In the prisoners dilemma game defective strategy always dominates the cooperative one since defection is always more beneficial independently on the strategy chosen by the opponent. In contrast, in the harmony game cooperation always dominates defection because the cooperation is more beneficial choice independently on the choice of the opponent. In the hawk-dove game it is beneficial to choose the action which differs from the action chosen by the opponent. This leads to the convergence case. It is more beneficial to be a hawk in a population in which the doves are in the vast majority. If the vast majority of population is hawks, it is more beneficial to be a dove. In the case of the stag-hunt game we have the divergence case. If agents of a certain type are in vast majority, the portion of the agents of another type will decrease and they will become extinct.

B. Evolution with Eight 1-State Agents

We have considered agents which always choose the same action: defection or cooperation. We can generalize agents if we make their choice dependent on the previous action of the opponent. To define such an agent we need to specify how it should respond on the previous defection or cooperation of its opponent. Moreover, when the agents make the first move in the sequence of the interactions no previous move of the opponent is available. So, we have also to specify what should be the first move of the agent. There are only 8 types of agent of this kind: ccc, ccd, cdc, cdd, dcc, dcd, ddc, and ddd. In our notation the first letter (c or d) denotes the action of the agent in the first move (cooperation and defection, respectively). The second and third letters denote the reaction of the agent on the cooperative and defective moves of the opponent in the previous game, respectively.

For 4 different games we have run evolution more that 8200 times starting from random proportions for the 8 different types of agents. In more detail, for every type of the agent we have generated a random number using a uniform distribution between 0 and 1. These numbers were normalized such that their sum is equal to 1. We used these numbers as initial proportions of the agents of different kinds.

We have found out that in the case of the prisoners dilemma the evolutions converged to one of the two states. The average payoffs of the agents in the two states are 0 and 2. The first state has been observed in 99.94% of cases and the second state has been observed in the rest of the cases. The observed states can also be distinguished by the proportions of the agents of different types.

In the case of the hawk-dove game also two convergence points have been identified. In the first and the second state the average payoff of the agents was equal to 1.15 and 1.93, respectively. The first state has been observed in 88.7% of cases.

In the case of the harmony game only one final state of the evolution has been identified. The average payoff of the agents in this state is equal to 2.

In the case of the stag-hunt game 5 different final states have been identified. The average payoffs of the agents in these states are: 0.00, 0.66, 1.00, 1.33 and 2.00. The frequencies of theses state are 10.47%, 6.70%, 17.66%, 0.35% and 64.82%, respectively.

C. Dynamics of Average Fitness

It is interesting to consider ability of the system, as a whole, to adapt to the external environment (rules of the game) and extract as much as possible from the environment (in terms of the average payoff of the agents). In other words, the system, as a whole, gets some payoff depending on how the agents (parts of the system) interact with each other and what are the relative portions different types of agents in the system. The system has a mechanism of changing its state (proportion of different kinds of agents) depending on the condition of the external environment. The question that we want to answer is if this mechanism gives the system an ability to adapt positively to the environment and benefit from its conditions.

As a result of the evolution the agents with low fitness become extinct. Does it mean that the average fitness of agents will increase over time as it is prescribed by the Fishers fundamental theorem of natural selection? In many cases the answer is negative. The classical example of the situation when natural selection constantly reduces the average fitness of the population is given by the system in which agents play with each other a single-shot Prisoners dilemma. In this case a population of only cooperators has

 ISBN: 978-1-61208-219-6

the highest average fitness, whereas a population of only defectors has the lowest. However, on the individual level, defection is always more beneficial than cooperation and, as a consequence, selections act to increase the relative portion of defectors. After some time, cooperators vanish from the population completely.

This example clearly demonstrate that the system is not always able to adapt positively to the conditions of the environment. However, as we have demonstrated earlier, if we enrich the system by more complex agents, it can exhibit properties of a positive adaptation. For example, in the case of the prisoners dilemma we have found a state in which the average payoff of agents in the system was equal to maximal one. However, this state was very rare. It has been reached only in 0.06% of cases, if we stared the evolution from random populations of agents of different kinds.

D. Stability of Adaptive Properties

The next question that we wanted to answer is how stable is the state in which the system is able to adapt positively to the priosner's dilemma conditions. To answer this question we started from the higher-payoff state of the prisoners dilemma and have performed a series of jumps to the conditions corresponding to the other types of games. Two sequences of conditions have been considered. The first sequence was: stag-hunt, hawk dove, prisoners dilemma. The second sequence was: hawk dove, stag-hunt, prisoners dilemma. In more detail, first we performed 700 hundreds steps of the evolution under the stag-hunt conditions. This number of steps was sufficient to reach a convergence. After that another 700 hundreds steps of evolution have been performed under the hawk-dove conditions. And finally we performed another 700 steps of evolution in the prisoners dilemma settings to check if the system is able to go back to the original higher-payoff state. In other words, we wanted to find out if evolution under other conditions destroys the ability of the system to adapt positively to the prisoners dilemma settings. To make the test even harder, we have performed a small randomization of the populations between the switching of the conditions of the external environment. The randomization was also performed to include into the evolutionary process under new conditions those types of agents which became completely extinct under the previous phase of the evolution. This test has shown that in 100 out of 100 cases the system was able to keep the ability to adapt positively to the prisoners dilemma. In other words, in spite on the fact that evolution under stag-hunt and hawk-dove conditions as well as the randomization changed the initial proportions of the agents the final evolution of the system, under the prisoners dilemma settings, was always able to converge to the higher-payoff state. Moreover, we found out that the ability to adapt positively to the prisoners dilemma conditions leads to the ability to adapt positively to the stag-hunt and hawk-dove conditions. In the stag-hunt and hawk-dove cases the evolution of the system converged to the highest-payoff state. An example of the positive adaptation is shown in the Figure 3. The red and black dots give an example of a positive and negative adaptation, respectively. The green lines show the allowed average payoffs in the converged state of the corresponding conditions. The test with the changed order of the stag-hunt and hawk-dove games leads us to the same conclusions as in the previous case.

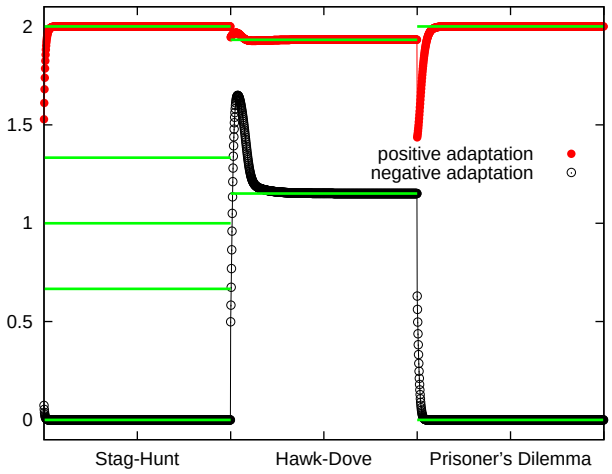


Figure 3. Examples of positive and negative adaptation of the system while changing conditions of the environment

Another interesting property of the considered system is that evolution of the system under new conditions does not completely destroy memory of the system about previous conditions of the evolution. This property can be proven in the following way. If we start from the populations corresponding to the higher-payoff state of the prisoners dilemma and evolve the system in the stag-hunt conditions until the convergence we will end up in the highest-payoff state. If we then evolve this system in the prisoners dilemma settings we will always converge back to the original higher-payoff state of the prisoners dilemma. We have observed this behavior in 100 cases out of 100. In contrast, if start from the random populations and evolve the system in the stag-hunt conditions in some cases we will converge to the higher-payoff state. Then, if we start from the higher-payoff state obtained in this way and start the evolution in the prisoners dilemma settings, we will converge to the higher-payoff state of the prisoners dilemma only in about half of the cases (48 cases out of 100 in our experiment). Thus we can say that the probability that evolution under the prisoners dilemma conditions will converge to the higher-payoff state depends not only on from what state we started the evolution (in our case the higher-payoff state of the stag-hunt conditions) but also on how the initial state was obtained. Despite the fact that in both cases we started the evolution from the higher-payoff state of the stag-hunt conditions, the system had much

 ISBN: 978-1-61208-219-6

more larger chances to converge to the higher-payoff state of the prisoners dilemma, if the system already was in the higher-payoff state of the prisoners dilemma game. This kind of memory is explained by the fact that a fixed state does not necessarily mean fixed populations of the agents of different types. In other words different populations of the agents of different kinds can correspond to the same state. For example, in the above considered example of the memory we had the higher-payoff state for the stag-hunt conditions. In this case only ccc and ccd agents are present in the system. However, the relative proportion of these two types of agents is not fixed. This degree of freedom can be used to encode the memory of the system.

IV. Conclusion

In this work we have presented a model of a complex adaptive system based on concepts of game theory. An evolving system of heterogeneous agents interacting with each other in a game theoretic way has bas introduced. The influence of the external environment has been modeled by payoff matrix of the game. The state of the system is given by the proportion of agents of different types in the population. The adaptive properties of the system have been introduced through an evolutionary selection of the agents. An ability of the system to increase an average payoff under given external conditions has been studied as an adaptive process. In this context the importance of the initial state of the system has been demonstrated. The stability of the positive adaptation has been considered. Moreover, it has been demonstrated that the introduced system could have a memory about the history of the changes of the external conditions. It has been shown that some memory about old conditions remains even if the system evolved in new conditions long enough to reach a convergence.

Acknowledgment

The research reported in this paper is supported by NWO (De Nederlandse Organisatie voor Wetenschappelijk Onderzoek) User Support Program Space Research. The project number is ALW-GO-MG/07-13.

References

[1] M. Hadzikadic, T. Carmichael, and C. Curtin, "Complex adaptive systems and game theory: An unlikely union," *Complexity*, vol. 16, pp. 34–42, 2010.

[2] J. M. Smith and G. R. Price, "The logic of animal conflict," *Nature*, vol. 246, pp. 15–18, 1973.

[3] A. Rapoport and A. M. Chammah, "The game of chicken," *American Behavioral Scientist*, vol. 10, pp. 10–28, 1966.

[4] M. M. Flood, "Some experimental games," *Management Science*, vol. 5, pp. 5–26, 1958.

[5] M. Dresher, *The Mathematics of Games of Strategy: Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ., 1961.

[6] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, pp. 1390–1396, 1981.

[7] A. Rapoport and A. M. Chammah, *Prisoner's Dilemma*. University of Michigan Press, 1965.

[8] B. Skyrms, *The Stag Hunt and Evolution of Social Structure*. Cambridge: Cambridge University Press., 2004.

[9] M. A. Nowak, "Five rules for the evolution of cooperation," *Science*, vol. 314, pp. 1560–1563, 2006.

[10] R. J. Aumann, "Acceptable points in general cooperative n-person games," *Annals of Mathematics Studies*, vol. 40, pp. 287–324, 1959.

[11] M. A. Nowak, S. Bonhoeffer, and R. M. May, "Spatial games and the maintenance of cooperation," *Proceedings of the National Academy of Sciences*, vol. 91, pp. 4877–4881, 1994.

 ISBN: 978-1-61208-219-6

Towards Formal Specification of Autonomic Control Systems

Elena Troubitsyna
Åbo Akademi University, Dept. of IT
Joukhaisenkatu 3-5A, 20520, Turku, Finland
Turku, Finland
e-mail: Elena.Troubitsyna@abo.fi

***Abstract*— Autonomic systems represent the next generation of software-intensive systems that merge software, computing, communication, sensing and actuating to create intelligent self-aware computing environment. Autonomic systems penetrate majority of critical infrastructures be it air-, rail and road traffic or power supply management. So far, little attention has been paid to theory and techniques for ensuring safety and resilience of such systems. Are autonomic systems to bring benefits or devastating hazards? To ensure harmless deployment of autonomic systems in critical infrastructures we should significantly advance our understanding of principles governing adaptive behaviour of such systems. Therefore, it is important to create formal techniques for modelling adaptive behaviour of autonomic control systems. In this paper, we discuss issues in modelling autonomic control systems that achieve self-adaptation through feedback loops and derive general guidelines for their formal specification.**

Keywords-*autonomic computing; control systems; action systems; formal specification*

I. INTRODUCTION

Autonomic systems are software-intensive systems that besides providing its intended functionality are also capable to diagnose and recover from errors caused either by external faults or unforeseen state of environment in which the system is operating. Autonomic systems are typical examples of self-adaptive systems. The concept of autonomic systems has been introduced in the recognition of complexity crisis. Currently the level of complexity of software has reached unprecedented level and we are no longer can reliably guarantee correct function of the system. Even though complexity is perceived as a major threat to dependability, self-adaptive systems are becoming more and more widely used in critical infrastructures. It is threatening situation that might cause catastrophic consequences.

Originally, autonomic computing paradigm was proposed in a very radical way: autonomic systems were supposed to mimic self-adaptive living organisms that can autonomously take care of themselves. In this paper, we are taking a stand that in the domain of critical systems we should take more moderate view and consider autonomic behavior that converges to a formally verified model that guarantees that the essential properties of the system are preserved despite self-adaptation.

In this paper, we discuss the principles of structuring formal models of autonomic control systems. We demonstrate how to formally specify behaviour of autonomic control system in the action systems formalism [2,3]. The formalism provides us with a unifying framework for developing terminating as well as reactive distributed systems. Our main development technique is stepwise refinement [4]. While developing a system by refinement, we start from an abstract specification and refine it into an executable program in a number of correctness preserving steps – refinements. Stepwise refinement allows us to incorporate system requirements into the specification gradually and eventually arrive at system implementation, which is correct by construction.

In this paper, we propose a general pattern for abstract specification and refinement of autonomic control systems. We present a novel pattern for an abstract specification of autonomic manager -- a components that is responsible for monitoring and adaptation of the control system. Our refinement steps gradually introduce detailed representation of data structures required to model autonomic system with a feedback control loop.

The proposed approach provides the developers with a rigorous framework for systematic development of fault tolerant distributed systems.

The paper is structured as follows: in Section II we describe a general architecture of the autonomic control systems with feedback loop. In Section III we present our formal modelling framework – the Action Systems formalism. In Section IV we demonstrate how to specify the autonomic manager and components of the autonomic control systems. Finally, in Section V we discuss the proposed approach and future work as well as overview the related work.

II. AUTONOMIC CONTROL SYSTEMS

The complexity of modern software systems and volatile environment in which they operate require novel computing paradigms to ensure that the system delivers the desired behaviour, i.e., is capable to adapt to the changing operating conditions. The autonomic computing paradigm is a promising research direction that puts the main emphasis on system self adaptation capabilities. Essentially, self-

 ISBN: 978-1-61208-219-6

adaptation is a capability of the system to adjust its behaviour without any human intervention.

In this paper, we consider issues in modelling systems that achieve self-adaption through feedback loops. In particular we focus on studying autonomic control systems. In general, a control system is a reactive system with two main entities: a plant and a controller. The plant behaviour evolves according to the involved physical processes and the control signals provided by the controller. The controller monitors the behaviour of the plant and adjusts it to provide intended functionality and maintain safety. The control systems are usually cyclic, i.e., at periodic intervals they get input from sensors, process it and output the new values to the actuators. The general structure of a control system is shown in Fig. 1.

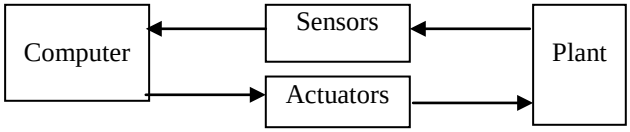


Figure 1. A general structure of a control system

A general structure of an autonomous control system is shown in Fig. 2.

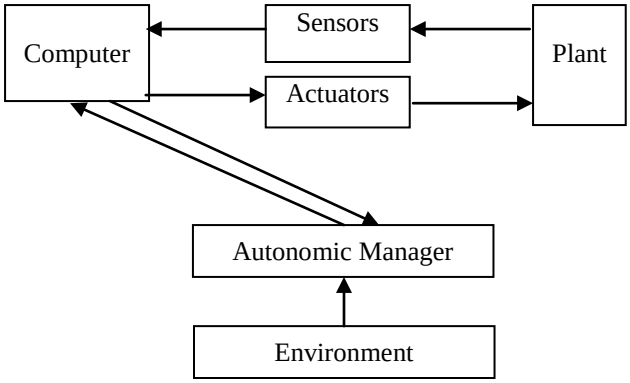


Figure 2. A general structure of an autonomic control system

A self-adaptive control system has an additional feedback control loop – we call it autonomic control loop. The loop has four main functions: monitor, analyse, decide and act. The monitoring activities are implemented via external sensors or monitors that collect data from the system and its environment. Usually the data acquired in the process of monitoring are filtered and stored in a log. The aim of collecting the data is to obtain an accurate model of the system dynamics and its current state. The collected data form the basis for diagnostics of failures, trends in operating environment, etc. There is a large variety of methods used for the analysis of the collected data. There are two general approaches: the first group relies on a reference model – a model of the expected system that is encoded into the analysis procedure at the design phase. Another type of analysis relies on inferring the model of the behaviour at the run time, i.e, no predefined model is given and the system is gradually building the model. In our paper we focus on the modelling the former class of systems.

Once the analysis of the collected data completes the planning phase takes place. The system decides on the strategy along which to continue its function. This strategy is then transformed into the control signals that are communicated to the actuators to implement the chosen strategy. This completed the cycle of the autonomic control loop.

In this paper, we focus on the analysis of autonomic control systems with a centralized autonomic manager. The autonomic manager is responsible for executing autonomic control loop. By communicating with the components of the system it collects the data required for diagnostics of the internal system state. The queues of the internal service requests as well as environmental conditions are monitored to define the usage profile and plan how the system functioning should proceed. The autonomic manager periodically sends diagnostics requests to the system components as well as requests reading from the external monitoring sensors and monitors the external service requests. This information is input to the analyzing component of the autonomic manager. Essentially the analyzing component compares the obtained data with the reference model and passes the control to the planning component. The planning component decides on the further strategy. The developed strategy is passed to the actuating component that sends the required control signals. The static view on the architecture of an autonomic manager is given in Fig. 3.

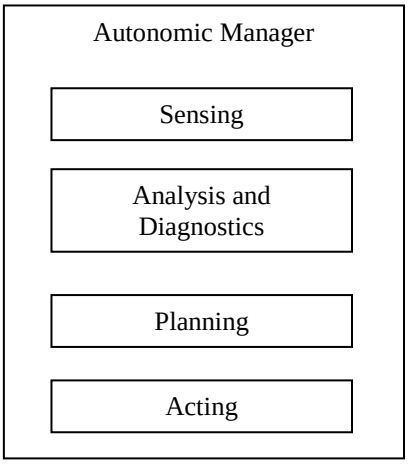


Figure 3.Structure of autonomic manager

As an example of an autonomic control system, let us consider an autonomic robot. Service robots form a quickly growing commercial area as well as research field. Service robots are designed to assist humans in performing services semi or completely automatically. There is a large variety of robots that are used for inspection, housekeeping, office automation and assisting elderly people or people with disabilities. The example that we present in this paper is

 ISBN: 978-1-61208-219-6

inspired by the intelligent service robot developed to assist elderly people. The robot should be able recognize a voice command and bring a desired object (e.g., medicine) from a certain position. We focus on the function of autonomous navigation. A user can command the robot to move to a specific position in the map to perform some task. For instance, the robot can navigate to its destination in the home environment via its sensors, which include laser scanners and ultrasonic sensors. The robot plans a path to the specified position, executes this path, and modifies it as necessary for avoiding obstacles. While the robot is moving, its constantly checks the data from its sensors.

Obviously, despite the complexity the robot should guarantee a high degree of dependability. For instance, we should ensure that the robot does not collide to the obstacles (and gets broken as a consequence leaving the person without the assistance). To facilitate design of dependable autonomous systems we propose to rely on formal modelling that provides us with a rigorous basis for reasoning about system behavior.

III. ACTION SYSTEMS

The action systems formalism [1] is a state-based approach to formal specification and development of parallel and distributed systems. The formalism has proven its worth in the design of complex parallel, distributed and reactive systems [5,13,14]. Below, we briefly describe the action systems.

A. Action Systems

The action system **A** is a set of actions operating on local and global variables:

$$\begin{aligned}\mathbf{A} :: \ & |[\ \mathbf{proc}\ p_1{*}=P_1;\ \ldots;\ p_N{*}=P_N;\ \ q_1=Q_1;\ \ldots;\ q_M=Q_M; \\ & \mathbf{var}\ v{*},u \bullet Init; \\ & \mathbf{do}\ A_1\ []\ldots[]\ A_K\ \mathbf{od}\ ]| : z\end{aligned}$$

The system **A** describes a computation, in which local variables u and exported global variables v^* are first created and initialised in *Init*. Then, repeatedly, any of the enabled actions *A1*, …, *An* is non-deterministically selected for execution. The computation terminates if no action is enabled, otherwise it continues infinitely. The actions operating on disjoint sets of variables can be executed in any order or in parallel.

The local variables u are only referenced locally in **A**, while the exported global variables v^* also can be referenced by other action systems. The imported global variables z are mentioned in the actions $A_1,\ldots, A_K$ but not declared locally. The identifiers of local, global imported and global exported variables are assumed to be distinct.

A procedure declaration $p=P$ consists of the *procedure header p* and the *procedure body P*. The procedures marked with * are declared as the exported procedures. They can be called from **A** and other action systems. The procedures $q_1,\ldots, q_M$ are the local procedures. They can be called only by **A**. The local and exported procedures are all assumed to be distinct.

The action A is a statement of the form $g(A) \rightarrow s(A)$, where $g(A)$ is a predicate over state variables (*the guard* of A) and $s(A)$ is a statement of Dijkstra's language of guarded commands [7] (*the body* of A). The action that establishes any postcondition is said to be miraculous. We take the view that an action is only enabled in those states in which it behaves non-miraculously. The guard of the action characterizes those states for which the action is enabled:

$$g(A) = \neg wp(A, false)$$

The actions are assumed to be *atomic*, meaning that only their input-output behaviour is of interest. They can be arbitrary sequential statements. Their behaviour can therefore be described by the weakest precondition predicate transformer of Dijkstra [7]. In addition to the statements considered by Dijkstra, we use non-deterministic choice A [] B between statements A and B, simultaneous execution of statements $A \| B$ provided A and B do not share state variables and prioritizing composition, $A//B$. Note, that the prioritizing composition selects the first action, if it is enabled, otherwise the second (the choice being deterministic):

$$A//B = A\ [] \ (\neg g(A) \rightarrow B)$$

The detail description of these operators can be found elsewhere [3,11].

The procedure bodies and the actions may contain procedure calls. As a parameter passing mechanisms we consider *call-by-value* denoted $p($**val** $x)$, *call-by-result* denoted $p($**res** $x)$ and *call-by-value-result* denoted p(**valres** x), where x stands for the formal parameters. We assume that the procedures are not recursive. An extensive study of procedures in the action system formalism has been conducted elsewhere [12].

B. Refinement

The main development technique for the action systems is stepwise refinement [2,3,4]. The action A is refined by the action C, written $A \leq C$, if, whenever A establishes a certain postcondition, so does C:

$$A \leq C \text{ iff for all } p:\ wp(A,p) \Rightarrow wp(C,p)$$

A variation of refinement is if A is (data-) refined by C via the relation R, written $A \leq_R C$. For this, assume A operates on the variables a,u and C operates on the variables c,u. The data refinement is defined as follows [2,3]:

$$A \leq_R C \text{ iff for all } p:\ R \wedge wp(A,p) \Rightarrow wp(C, (\exists a \bullet R \wedge p)) \qquad (2)$$

Data refinement allows us to replace the variables in the actions. The relation R defines the correspondence between the replaced variables a and the newly introduced variables c.

 ISBN: 978-1-61208-219-6

When carrying out refinement in practice, one seldom appeals to the definitions (1) and (2). Instead certain pre-proven refinement rules are used.). Instead certain pre-proven refinement rules are used. For instance, *Rule 1* and *Rule 2* below are examples of derived rules for verifying refinement between actions and action systems. Let assume that *A* operates on variables *a,z* and *C* operates on variables *c,z.* Let *R* be the relation over *a,c,z:*

Rule 1: $A \preceq_R C$ iff

(i) $R \wedge g(C) \Rightarrow g(A)$

(ii) $\forall p.(R \wedge g(C) \wedge wp(sA,p) \Rightarrow wp(sC, \exists a \bullet R \wedge p))$

Rule 2: For action systems ***A*** and ***C***, $\boldsymbol{A} \preceq_R \boldsymbol{C}$, iff

(i) Initialization: $C0 \Rightarrow (\exists a \bullet R \wedge C0)$,

(ii) Actions: $Ai \preceq_R Ci$ for all *i*,

(iii) Exit condition: $R \wedge (\vee_i g(Ai)) \Rightarrow (\vee_i g(Ci))$

The proofs of these rules can be found elsewhere [2,3].

The action system formalism has been successfully used in component-based design. The formalism supports three most important modularization mechanisms: procedures, parallel composition and data encapsulation [3,12]. The components specified as action systems can communicate via shared variables, shared actions or remote procedure calls. Let us consider two action systems **A** and **B**

A :: |[**proc** q_1*=Q_1; … q_N*=Q_M;
$pA_1=PA_1$; … $pA_M=PA_N$;
var v*,a • InitA**;**
do A_1 []…[] A_K **od**]| : *z*

B :: |[**proc** r_1*=R_1; … r_S*=R_S;
$pB_1=PB_1$; … $pB_T=PB_T$;
var *w*,b • InitB**;**
do B_1 []…[] B_L **od**]| : *y*

where *v** and *w**, *a* and *b*, *z* and *y* are pairwise distinct. Moreover, the local procedures pA_1, … pA_M, pB_1, … pB_T declared in **A** and **B** are distinct too. The *parallel composition* **A**||**B** of **A** and **B** is the action system **C**

C :: |[**proc** q_1*=Q_1; … q_N*=Q_M;
r_1*=R_1; … r_S*=R_S;
$pA_1=PA_1$; … $pA_M=PA_N$;
$pB_1=PB_1$; … $pB_T=PB_T$
var *v*, w*,a, b • InitA||InitB**;**
do A_1 []…[] A_K [] B_1 []…[] B_L **od**]| : $z \cup y$

Hence, the parallel composition combines the state spaces of the constituent action systems, merging the global variables and global procedures and keeping the local variables distinct. The prioritizing composition of action systems is defined similarly. However, in the resultant action system **A//B** the preferences are given to the actions of the action system **A.** The refinement of component systems is usually carried out by application of the following refinement rules:

Rule 3: $\boldsymbol{A1||A2} \preceq_R \boldsymbol{C1||C2}$ if $\boldsymbol{A1} \preceq_R \boldsymbol{C1}$ and $\boldsymbol{A2} \preceq_R \boldsymbol{C2}$

Rule 4: $\boldsymbol{A1//A2} \preceq_R \boldsymbol{C1//C2}$ if $\boldsymbol{A1} \preceq_R \boldsymbol{C1}$ and $\boldsymbol{A2} \preceq_R \boldsymbol{C2}$ and

$R \wedge g(\boldsymbol{A1}) \Rightarrow g(\boldsymbol{C1})$

The action systems and refinement provide us with a suitable framework for formal specification and verification of the behaviour of autonomic control systems. The aim of this paper is to propose patterns for modelling autonomic control systems that rely of a feedback loop for their self-adaptation.

IV. SPECIFYING AUTONOMIC CONTROL SYSTEM AS ACTION SYSTEMS

A. Autonomic component

We start deriving a formal specification of an autonomic control system from defining a structure of an autonomic manager. Our specification follows the architecture depicted in Fig. 3. Essentially, it can be seen as a sequential composition of the actions modelling data collection, data analysis and error diagnostics, planning of the next control step and setting actuators accordingly. Formally, we define it as follows:

AM :: |[const
eval: DATA x DATA x DATA x STATE
planning : STATE x DATA x PLAN
acting: PLAN x A_STATE
next_exp_state: STATE x A_STATE x DATA
var *int_data, ext_data, ref_data: DATA*
cur_state :STATE
cur_plan :PLAN
act_state:A_STATE
flag: {sen,an,pl,act}•
INIT *int_data, ext_data, ref_data:: DATA*
cur_state := init_state
cur_plan := nil_PLAN
act_state:=idle
*flag: =sen***;**
do
flag=sen -> int_data, ext_data:: DATA || flag:= an
[] *flag=an ->*
cur_state:= eval(int_data, ext_data,ref_data)
|| flag:=pl
[] *flag=pl ->*
cur_plan := planning(ext_data,cur_state)
|| flag:= ac
[] *flag=ac->*
act_state := acting(cur_plan)
|| ref_data:=next_exp_state(cur_state,act_state)
|| flag:= ac
od]|

In the **cons**t clause, we have defined a number of abstract functions. The function *eval* models the analysis of data. Essentially, it compares the data obtained from the internal and external sensors with the reference model. The function *planning* takes as an input the data obtained from the external sensors and the current system state to decide on the next system behaviour. The behaviour is modelled by an abstract set *PLAN*. The function *acting* defines the new states of actuators based on the defined plan and their current state. Finally, the function *next_exp_state* defines the next value of reference model against which the system state will be compared.

In the **var** clause, we have defined the variables of the model. The variables *int_data* and *ext_data* represent readings of internal and external sensors correspondingly. The variable *ref_data* defines the next expected value of the reference model. The variable *cur_state* defines the current state of the system. Finally, the variable *act_state* abstractly models the state of the actuators. The variable *flag* is an auxiliary variable modelling the progress of system execution.

Let us now describe the actions modelling the behaviour of the autonomic component. The first action models the results of monitoring via the external and internal sensors. The internal sensors supply the information regarding the state of the system components. This information can be used to detect occurrence of failure and deviations from the expected behaviour. The external sensors bring the fresh information about the operating environment of the system. For instance, in the case of the autonomic robot, the external sensors will signal about the obstacles detected on route. These data are compared against the reference data. The general mechanism used for the comparison is to ensure that the detected system state matches the expected system state. Based on that comparison, the variable *cur_state* obtains the new value. The next action relies on the results of the analysis to define the current plan. In general, a plan corresponds to a certain mode of the components. As soon as the plan is defined, the new states of the actuators can be computed. By relying on the assigned states of the actuators, we can also compute the expected value of the reference model. This value will be used in the next cycle to diagnose the state of the system.

The actions are executed cyclically. At each iteration, the autonomic manager repeats the same sequence of actions, as defined by our model *AM*.

B. Specification of the autonomic component

Next, we demonstrate how to specify a self-aware component. A self-aware component besides providing its intended functionality also detects errors in its own functioning and raises corresponding exceptions, if the requested service cannot be executed. The self-awareness capabilities not only allow us to build the system in a well-structured way but also enable an effective fault tolerance.

In our initial specification, we assume that functioning modes of the autonomic components are transparent to the entire system. Such an abstraction allows us to significantly simplify the initial specification. The real communication mechanism is introduced at the consequent refinement steps.

When a component is not involved in executing a request sent by the autonomic manager, it is in the state *idle*. The autonomic manager may request a certain service by placing the corresponding data in *req_buffer*. If a component is idle and a request arrives then the component starts to execute this request and enters the state *executing*. Upon completing the requested service the component becomes idle again.

Therefore, the autonomic component **AC** can be specified as follows:

AC :: |[**var** *req_buffer*: Buffer*
resp_buffer:Buffer*
int_exc:Exceptions
state:{idle,executing,failed}
do N || F od]|

where

N :: |[**do** *state=idle /\ req_buffer ≠empty /\ no_exc →*
state:= executing || req_buffer:=tail(req_buffer)
[] $_{i: 1..N}$ *state:= executing /*
no_exc /\ event_i →
reaction_i
[] int_exc := exc
[] reaction_i ||
state:= idle||
resp_buffer:= resp_buffer^result
od]|

F::|[**do**[]$_{j:1..M}$*int_exc→internal_exc_handling||*
rec::{OK,Failed}
[] $_{j:1..M}$ *int_exc/\ rec=OK→ int_exc:= Nil*
[] $_{j:1..K}$ *int_exc/\ rec=Failed→ state:=failed*
[] **od]|**

The action system **N** defines the intended functionality of the autonomic component, i.e., executing requests that are sent to it by the autonomic manager. When a request is chosen for execution, it is deleted from the buffer of requests *req_buffer*. When the autonomic component successfully completes request execution, the result is put into the response buffer *resp_buffer*. The action system **N** models successful service provisioning

Upon detecting an error, an exception *int_exc* is raised and control is passed to the subsystem **F**. The action system **F** specifies handling of errors. If error handling succeeds then the component resumes its normal function, i.e., control is passed to the subsystem **N**. However, if error handling fails, the component state is changed to *failed*.

 ISBN: 978-1-61208-219-6

B. Specification of the overall architecture

We aim at defining the overall system architecture by composing models of components of control systems with the model of autonomic manager. To enforce separation of concerns we introduce self-awareness capabilities into the model of system components.

Assume that $\mathbf{AC} = \{\mathbf{AC_0}, \ldots, \mathbf{AC_M}\}$ is a set of autonomic components. We specify an autonomic control system as a prioritizing composition of the autonomic manager and the action systems specifying the autonomic components:

$$\mathbf{S:: AM//AC_0 \,||\, AC_1 \,||\ldots||\, AC_M}$$

Initially, the components communicate via shared variables. In the refinement process, we replace shared variables by the remote procedures. Further refinement steps (omitted here) allow us eventually decompose the system into independent components.

The proof-based verification associated with the refinement allows us to reason about preservation of important system properties, such as correctness, safety and fault tolerance in presence of self-adaptation. Moreover, by instantiating the abstract functions with their concrete counterpart, we arrive at the additional properties ensuring correct execution of self-adaptation scenarios. In the large distributed systems ensuring these properties at the architectural level brings benefits of early problem discovery. Hence, it allows us to arrive at a more robust design and speed up the development cycle.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed guidelines for structuring formal models of autonomic control systems based on feedback loops. The approach is based on the formal derivation of correct by construction system via stepwise refinement. We demonstrated how to derive system architecture in a formal way and structure specifications of self-aware components and autonomic manager.

A variety of approaches has been proposed to model autonomic control systems (e.g., see [4,5] for review). However, majority of these approaches focus on modeling details of self-adaptation mechanisms or their implementation details. In our work, we aimed at defining high-level architectural guidelines for modeling autonomic systems.

Sensoria project [8] has significantly advanced the area of formal modelling of adaptive systems. It has developed a number of modelling and programming primitives for just-in-time composition. Moreover, mathematical models for reasoning about correctness as well as quantitative analysis have been proposed as well. However, the project mainly aimed at modelling services and service-oriented systems rather than autonomic control systems – the goal that we have pursued in this paper.

As a future work, we aim at validating the proposed approach by a number of case studies. Moreover, it would be interesting to develop patterns for knowledge collection and representation in the formal model of autonomic control systems.

REFERENCES

[1] T. Anderson and P.A. Lee. Fault Tolerance: Principles and Practice. Prentice-Hall, Englewood Cliffs, 1981.

[2] R.J.R. Back, "Refinement calculus, Part II: Parallel and reactive programs". In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg (Eds.), *Stepwise Refinement of Distributed Systems*, pp. 67-93. New York, Springer-Verlag, 1990.

[3] R.J.R. Back and K. Sere, "From modular systems to action systems", *Software –Concept and Tools 17*, pp. 26-39, 1996.

[4] R.J.R. Back and J. von Wright, *Refinement Calculus: A Systematic Introduction*. New York, Springer-Verlag, 1998.

[5] M. Butler, E. Sekerinski, and K. Sere, "An Action System Approach to the Steam Boiler Problem", In J.-R. Abrial, E. Borger and H. Langmaack (Eds.), *Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control*, pp. 129-148, New York, Springer-Verlag, 1996.

[6] F. Cristian. "Exception Handling". In T. Anderson. *Dependability of Resilient Computers*, pp. 68–97. BSP, 1989.

[7] E. W. Dijkstra, *A Discipline of Programming.* Englewood Cliffs, NJ: Prentice Hall, 1976.

[8] EU FP6 Project Sensoria: Software Engineering for Service-Oriented Overlay Computer. http://www.sensoria-ist.eu/

[9] J.-C. Laprie, *Dependability: Basic Concepts and Terminology*. New York, Springer-Verlag, 1991.

[10] N.G. Leveson, *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[11] E. Sekerinski and K. Sere, "A Theory of Prioritizing Composition", *The Computer Journal*, *39*(8), pp. 701-712, 1996.

[12] K. Sere and M. Waldén, "Data Refinement of Remote Procedures", *Formal Aspects of Computing*, *12*(4), pp. 278–297, 2000.

[13] K. Sere and E. Troubitsyna, "Safety Analysis in Formal Specification", *Proc. World Congress on Formal Methods in the Development of Computing Systems*, pp. 1564-1583, Springer, 1999.

[14] E. Troubitsyna, "Developing Fault-Tolerant Control Systems Composed of Self-Checking Components in the Action Systems Formalism", *Proc. The Workshop on Formal Aspects of Component Software*, pp. 167-186, 2003.

 ISBN: 978-1-61208-219-6

PROTEUS: A Language for Adaptation Plans

Antinisca Di Marco, Francesco Gallo
University of L'Aquila
Department of Information Science
L'Aquila, Italy
antinisca.dimarco, francesco.gallo@univaq.it

Franco Raimondi
School of Engineering and Information Sciences
Middlesex University
London, UK
f.raimondi@mdx.ac.uk

***Abstract*—The purpose of this paper is to present PROTEUS, a new language and, more in general, an approach for the construction of reconfiguration plans to support adaptation in systems belonging to different domains. The approach allows the management of runtime adaptation, preventing that running shared services are terminated and taken off-line while being reconfigured, causing inefficiency and disruptions. We introduce the new concept of *virtual membrane*, in order to give the system ability to adapt itself at run time in front of a new reconfiguration plans. We apply PROTEUS on an example to show the expressiveness and power of the new language.**

Keywords-Adaptive Systems. Reconfigurations Rule. DSL. ADL.

I. Introduction

Modern software systems show a complexity and a size that can often make difficult their maintenance and adaptation to environmental changes. Moreover, there is an increasing need to satisfy functional and non-functional requirements when a system lives in an environment composed by independent and often competing entities, such as in the web service market [1][2]. The current trend is to delegate adaptation [3][4] and fault tolerance [5] to the system itself by means of redundancy and other techniques.

To support these capabilities, we propose a new reconfiguration language called PROTEUS, that aims at building and managing rules to reconfigure software applications. The new rules are generated and managed using a new concept: the Virtual Membrane. A virtual membrane defines one input for a new reconfiguration plan in which all the resources involved are subject to adaptation. The intuition is that rules represent views of a certain system and define a "new" configuration for the system which is compatible with the new state of the environment (i.e., the new context in which the system lives). Note that it is out of scope of this paper to define how the reconfiguration plans are created. For this aim, we envisage an engine (local to the system) able to interpret the context changes and to define the suitable reconfiguration plan (expresses by means of PROTEUS) to adapt the system to the new situation. In our mind, such engine needs to be distributed across the system modules/resources. The rest of the paper is organized as follows: Section II describes PROTEUS and fundamental concepts related to it, using a scenario that describes the management of a WSN (Wireless Sensor Network) for the detection of a traffic jam. In Section III and IV, we formally introduce the main concepts of PROTEUS. Related work is presented in Section V. Section VI provides concluding remarks and future work.

II. Motivational Example

In this section, we introduce the Traffic Jam case study to which we apply some of the most significant actions of reconfiguration introduced by PROTEUS and the concept of virtual membrane. With this example, we want to show the ability of PROTEUS to manage the system resources at a high-level, and to introduce the concept of virtual membranes, which allows to separate the behavior from the system resources and organize them so as to have different independent views.

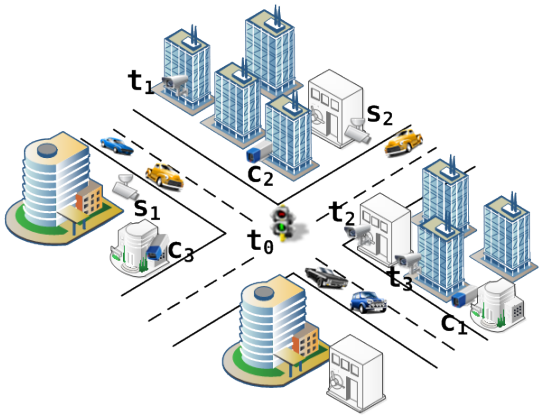


Figure 1. Scenario

The scenario of Figure 1 shows an urban environment, consisting of roads, buildings, cars and a traffic jam control system composing of a series of sensor nodes installed along the ways or on buildings. The system is composed of a set of wireless nodes of various types and one base station contained in the traffic light. The role of base stations is created at run-time by means of clustering algorithms [6], capable to elect a cluster head. In case of failure of the base

 ISBN: 978-1-61208-219-6

station, a new cluster head(base station) is elected among the nodes with sufficient computing power. As we will see in Section III-B, we identify the various components of the system with a set of *resources*, namely RES. For the sake of this example, RES is defined as follows:

$$RES = \{t_0, t_1, t_2, t_3, c_1, c_2, c_3, s_1, s_2\}$$

where:

- t_0 represents the base station/traffic light; it receives data from various camera nodes and turns on the traffic lights if necessary. It also has a greater computing power than other nodes on the network;
- t_i are cameras having the ability to rotate around their vertical axis. They can not change their angle of inclination and are equipped with temperature sensors and sensors to detect CO_2 levels;
- c_j are cameras having the ability to rotate around their vertical axis and to vary their angle of inclination.
- s_k are fixed cameras that can cover only a limited part of the road.

We assume that all nodes are powered by a solar panel, can communicate with their neighbour nodes, and the base station has two types of clients:

1) the system for the management of the traffic light that is activated when a traffic jam is detected;
2) the system for detection of CO_2 level. An alert is triggered when this level exceeds a given limit value of CO_2.

As mentioned above, only the nodes of type t_i have the capacity to measure the level of CO_2. We call this feature f_1, and we extend the set of resources RES with this additional feature:

$$RES = \{t_0, t_1, t_2, t_3, c_1, c_2, c_3, s_1, s_2, f_1\}$$

The two base station clients have a different logic view of the network, since they have different monitoring purposes. In our approach these two logic views can be synthesized by two *virtual membranes*:

$$v_1 = \{t_0, t_1, t_2, t_3, c_1, c_2, c_3, s_1, s_2\}$$

and

$$v_2 = \{t_0, t_1, t_2, t_3, f_1\}$$

In our formalism, virtual membranes are members of the set of resources, and thus

$$RES = \{t_0, t_1, t_2, t_3, c_1, c_2, c_3, s_1, s_2, f_1, v_1, v_2\}$$

Consider now the following scenario: due to a technical problem, some of the nodes belonging to the membrane v_1 start sending incorrect data to the base station t_0, causing improper behaviour of the traffic lights.

At this point we need to ensure that the base station does not continue to receive incorrect data from sensors in v_1, still guaranteeing the management of the traffic light in case a traffic jam is detected.

It is also necessary to select in an appropriate way the system resources that will be affected by the reconfiguration plan. We do this by defining a predicate P_1 defined as follows:

- $P_1 = \{r_i == v_1\}$

Intuitively, the predicate is true for a resource iff that resource is the virtual membrane v_1. We employ this predicate in PROTEUS to send a reconfiguration message that deactivates the v_1 virtual membrane; that is, it makes inactive the corresponding system view or behavior. The syntax for this reconfiguration plan is as follows:

$\{\forall\, r_i \in RES :\ P_1(r_i) = true)\} \Longrightarrow$
$program\,\{\ \ vInactive(self, vMembrane\, v_1);$
$\}$

We refer to [7] for the definition of the full syntax of PROTEUS. We want to recall by turning off v_1, the behavior implemented by it is inhibited. Whereas, inactivating v_1 does not inactivate or stop the resources belonging to it. This means that, if the resources used by v_1 are shared with other virtual membranes, the manipulation of v_1 is absolutely transparent to them since the shared resources continue to work for them. In our example, this means that all the resources in v_2 shared with v_1 continue to work even if v_1 is inactive.

Once the virtual membrane v_1 is inactive, to continue to provide the traffic light management in case a traffic jam occurs, we use membrane v_2: the excessive production of CO_2 in a particular point of a road network presumes a large number of combustion engines in transit, or a high density of vehicles not in motion, hence a traffic jam.

We can then use f_1 to enable the semaphore when the CO_2 level exceeds a certain threshold by means of the following property and reconfiguration plan:

- $P_2 = \{r_i == f_1\}$

$\{\forall\, r_i \in RES :\ P_2(r_i) = true)\} \Longrightarrow$
$program\,\{$
$vCreate\, v_3\{$
$v_3(P_2)\{$
$if(f_1 == threshold)\{active(self, t_0)$
$\}$
$if(f_1 < threshold)\{deactive(self, t_0)$
$\}$
$\}$

In this reconfiguration plan, a new virtual membrane is created (namely v_3) that is composed only by f_1. Note that the reconfiguration plan makes use of f_1 to turn on or off the traffic light depending on the CO_2 level. Indeed, f_1 is a feature implemented by the resources in v_2 hence v_3 uses implicitly the resources belonging to v_2. Even if v_2 and v_3 share implicitly and explicitly the same resources, they do not affect each other since the former is transparent to the latter and viceversa.

 ISBN: 978-1-61208-219-6

III. SETTING THE CONTEXT

In this section, we provide an abstract formalization of the concepts introduced in the previous section. An adaptive system is a system able to provide adaptation features, following the application of specific capabilities. In PROTEUS, an adaptive system is composed of two logical layers:

Application Layer, which represents the application logic of the system where all hardware and software features, and their relationships are defined and managed;

Adaptation Layer, providing features for reconfiguration and adaptation. In particular, this layer is able to process the reconfiguration specified by PROTEUS.

A. Event Concept

The adaptation is triggered by external or internal events; once the system has captured one of these events, it creates a virtual membrane that selectively aggregates the resources necessary to implement the plan of reconfiguration.

Formally, an event is a logical predicate, which can have the following forms:

$$\{\forall\, r_i \in RES:\ P(r_i) = true)\} \Longrightarrow reconf$$

or

$$\{\exists\, r_i \in RES:\ P(r_i) = true\} \Longrightarrow reconf$$

where P is a property from a $properties\_set$, which determines the resource set to select; $reconf$ is the new behaviour to be implemented. At the present, we assume that the reconfiguration plan is generated manually or by an external entity.

B. Resource Concept

In PROTEUS, the concept of resource plays a crucial role; it is an aggregation of *features* and *attributes*, as described in Figure 2. Both features and attributes are the resources on which it is possible to act directly.

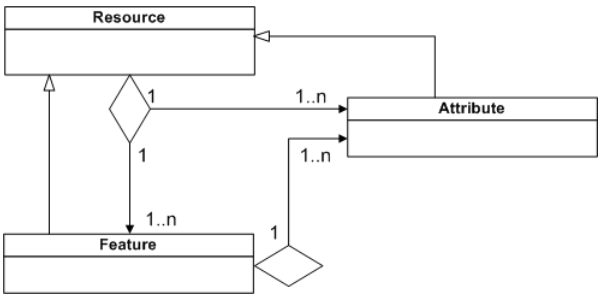


Figure 2. Resource concept

From a formal point of view, resources are assimilable to elements that belong to a *multiset* (the concept of *multiset* is a generalization of the concept of set where the multiplicity of any element could be greater than one.): this allows us to group together instances of various types (and possibly the same type), and act on them in a distributed mode. More formally, we identify resources with the set RES, where:

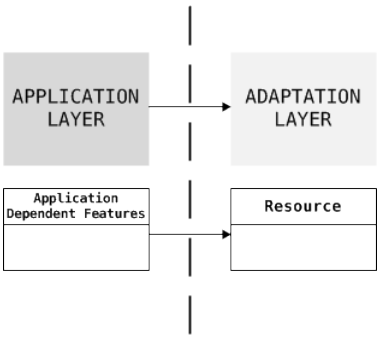


Figure 3. Resource and Feature

Definition: $RES = \{r_1, r_2, ..., r_n, vm_1, ..., vm_k\}$ is a *multiset* in which all the resources that are part of the application layer are collected. Each of these resources has a *status* that tells the user if it is *active* or *not active*. The multiset can contain both hardware and software resources, but also logical features represented by virtual membranes.

Whenever the system is involved in an adaptation, PROTEUS is able to add or remove resources, enable or disable existing features, etc..

Thanks to the concept of attribute, PROTEUS is able to further enhance the capabilities of a resource, allowing the user to change the state of the resource or activate components of the resource made temporarily silent, or added later (where possible). In this sense, we have:

application-depended features, that specialize the generic concept of feature resources, depending on the application domain and;

adaptation-depended features, that express the logic to adapt the features of the application layer. Figure 3 describes the concept of abstraction that PROTEUS introduces: it allows us to keep separate the application logic of the system from the logic of adaptation, ensuring the following advantages:

- the *adaptation layer* is generic, i.e., it is not bound in any way to the application layer. In this way the complexity of the application layer of the system is not weighed down by specific reconfiguration logic;
- the *application layer* can be customized according to the domain application, to support specific needs of adaptation.

C. Resource Type

Each resource has a *type* that characterizes and distinguishes it in the system. Our approach provides two kinds of *resource types*:

Adaptation Related resource types are the mechanism through which our approach allows to insert new behaviors

in an application, or update a resource or collection of resources. This corresponds, for example, to adding a method m_j to a class C_i or to deactivating a feature, or a part of it.

Application Related resource types are determined by the application domain. For example, consider the WSN domain described in Section II: an application type may be defined by a particular sensor, called $Sensor_j$.

IV. PROTEUS MAIN FEATURES

In this section, we present the main features of PROTEUS such as the adaptation actions (in Section IV-A), properties (in Section IV-B) virtual membrane (in Section IV-C) and the actions PROTEUS defines for it (in Section IV-D). The interested reader can refer to [7] for the complete syntax of PROTEUS.

A. Adaptation Action

In this section, we describe the principal actions that PROTEUS provides.

In Proteus a reconfiguration plans is a sequence of *adaptation action*s as summarized in Table I. These actions are described as follows:

- *add* allows the user to add some resource to the system. We recall that a resource can be: *vMembrane*, *feature*, *attribute* or *domain specific*. For example, if the application domain is a wireless sensor network, we may want to add to the network a new type of sensor, called $Sensor_j$;
- *bind* between resources. For example, considering a WSN, a bind may reflect the need to initiate a communication between a base station and a peripheral sensor;
- *remove* resources. For example, considering a WSN, remove a resource could mean that a node is unavaible due to failure;
- *update* resources. For example, this allows the user to update an old version of a software component, change the communication protocol between nodes on a network, etc.;
- *activate* resources. For example, it allows to activate one particular sensor of the WSN nodes;
- *deactivate* resources. For example, if a network of sensors needs to reduce energy consumption, the user can disable one or more of the sensing node modules.
- $domain\_specific$ defines domain specific adaptation actions the application developer wants to introduces in the language. For example, a developer could build reconfiguration actions that are a combination of those above in order to make atomic a sequence of them or reconfigure a WSN according to a certain policy.

B. Properties

For each resource identified, we associate a set of constraints required to change the current configuration of the system or subsystem. We can simultaneously select a set of resources to apply an adaptation plan consisting of a sequence of actions. The resources spaces (features) involved are selected through the use of *properties*, formally defined as:

Properties: $PRO_{j=\{1..m\}} = \{constraints^{+}\}$, is the set of properties that the system must satisfy. Each property is defined by one or more constraints, represented by logical predicates.

C. Virtual Membrane

In order to adapt a single resource or a pool of resources, we introduce, through PROTEUS, the concept of **Virtual Membrane** that provide tools that support the ability to adapt the system to internal or external events. A virtual membrane defines the boundaries of the portion of the application subject to adaptation. The purpose of a membrane is to select a resource or group of resources belonging to the system, to define new interactions within the system and, consequently, new behaviors, or to modify the behavior of the internal resources of the application.

The advantages of introducing the concept of virtual membranes are the following:

- the adaptation operations are performed at runtime, ensuring continuity of service;
- the creation of a virtual membranes allows the selection of a set of resources, generating a specific view of (the portion of) the system involved in the change. This view does not interfere with the various other views of the system since they are *behaviors*;
- the previous point has the consequence that the adaptation is completely transparent to the user. Each view gives a level of abstraction that is only accessible to the user/subsystem which are enabled, Figure 4;

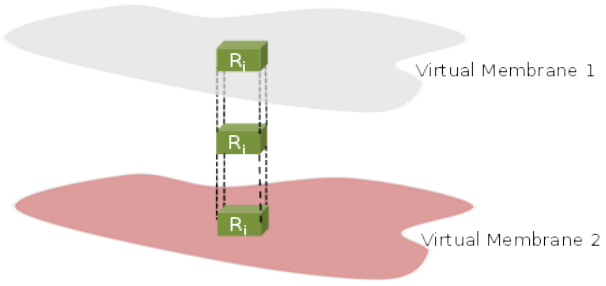


Figure 4. Virtual Membrane

- once created, the virtual membranes are comparable to system resources; in fact they can be manipulated through specific actions.

Definition:

Virtual Membrane: A virtual membrane $vm_j = \{r_i \in RES : PRO_j(r_i) = true, i = \{1..n\})\}_{j=\{1..m\}}$ is a set that contains all elements of the system that verify the property PRO_j.

Adaptation Action	Description
Add	The action allows the user to add some element to the system. The element type can be: *vMembrane*, *feature*, *attribute* or *domain specific*.
Bind	The action creates a bind between two system's resources.
Remove	The action allows the user to remove a resource from the system. The removal of a resource is final and it can not be used any longer.
Update	The action updates a system resource.
Activate	This action allows the user to enable existing resource.
Deactivate	This action allows the user to disable temporarily or permanently a resource.
+domain_specific	Developers can define their own action, depending on particular applications needs.

Table I
ADAPTATION ACTION

Virtual Membrane Sets: $VM = \{vm_j\}_{j=\{1..m\}}$ is the set of all virtual membranes generated in the system.

In PROTEUS virtual membranes are considered as standard resources: this is to be able to adapt the virtual membrane in a uniform way with others concepts. In this way, the virtual membrane is itself adaptable and it can evolve over time. To manage this type of resource we defined specific actions, called *Virtual Actions*, specified in the following section.

D. Virtual Action

PROTEUS specifies a set of actions to create and manipulate virtual membranes that are listed in Table 2. Recall that a virtual membrane is a logical resource of the system, and thus the actions reported affect the behavior associated with the virtual membrane, rather than the resources that implements it.

V. RELATED WORK

In this section, we summarize the characteristics of some architectural styles and frameworks useful to develop and manage reconfiguration in various types of software systems.

We discuss the state of the art by comparing it with PROTEUS. To this end we organize the presentation of the main features of the related work in Table III. Each row of the table is dedicated to an approach whereas the first row is related to PROTEUS and provides the reconfiguration actions of PROTEUS that we consider as the minimum set for the construction of a language able to define a reconfiguration plan.

The approaches described here are applicable to levels of granularity ranging from the reconfiguration of the architectural elements to objects in the actual running the system.

The various tools considered are as follows:

- **FScript** [8], a Domain-Specific Language used for reconfigurations of Fractal architectures.
  FScript introduces a new notation, called FPath, which is designed to express queries on a FRACTAL architecture and navigate it, selecting items on the basis of logical predicates. FScript provides access to all of the primitive actions present in Fractal reconfiguration, and enables the user to define customized reconfiguration.
- **FORMAware** [9], incorporates component-based development and architecture knowledge. Furthermore, this framework provides flexible mechanisms to recompose components at runtime to address scalability, mobility and general architecture evolutionary scenarios.
- **Dynamic Contextual Adaptation with a DSL** [10], defines high-level declarative constructs that can be used to specify the adaptation of the application behaviour to specific situations. The language is supported by a framework that enables the exchange and merge of behaviours on-the-fly.
- **DSL for ATRON robots** [11], is a role-based language that allows the programmer to define roles and behaviour for a physical module that is activated when the structural invariants associated with the role are fulfilled.
- **Representational state transfer (REST)** [12], is a paradigm for Web applications that allows the manipulation of resources using methods GET, POST, PUT and DELETE of the HTTP protocol. Basing its foundations on the HTTP protocol, the REST narrows its paradigm field of interest to applications that use this protocol to communicate with other systems.
- **CLOUD Computing** [13], refers to a collection of technologies that allow store/archive and/or data processing (via the CPU or software) typically in the form of a service offered by a provider to the customer, through the use of hardware/software on a distributed and virtualized network.

In Table III, we show the reconfiguration actions that the various frameworks provide. For simplicity, the names of the actions of reference used are those introduced by PROTEUS: in each corresponding cell there is a brief description of the semantics of the action in reference to the particular framework.

With regard to REST, the adaptation can be conceived as the ability of the system (web) to expose applications and features, such as services (web), in the form of callable API from a client. Through the use of various kinds of connectors we can define a large number of interactions between clients and resources, facilitating the system scalability and

 ISBN: 978-1-61208-219-6

Virtual Action	Description
vCreate	This action allows to **build** the membrane by means of a constructor that is capable of aggregating elements of the systems that satisfy the properties introduced by the event that triggered the adaptation.
vCompose	This action allows to **combine** two virtual membrane through a *compose operation* set. The operations are *union*, *difference* and *intersection*. The use of these operators are allowed in their standard sense because the system resource set is defined as multiset.
vRemove	This action allows to **remove** from the system the constraints that generated the specific virtual membrane.
vActive	This action allows to **enable** the specific behaviour implemented a virtual membrane. This action affects the *status field*.
vInactive	This action allows to **disable** the specific behaviour implemented a virtual membrane. This action affects the *status field*.

Table II
VIRTUAL ACTION

	Adaptive Action						
PROTEUS	Add	Bind	Remove	Update	Activated	Inactivated	Customization
FRACTAL	**ADL module**: - Composite components; - Instantiation a component from a ADL definition (Java code generation); - Shared components; - Content Controller (add sub component); - **Binding Controller**: - bind;	**Communication path between component interfaces**: - *Primitive Binding,* is a binding between one client interface and one server interface, in the same address space; - *Composite Binding,* is a communication path between an arbitrary number of component interfaces, of arbitrary language types;	**Content Controller**: - remove sub component; **Binding Controller**: - unbind;				**FScript, FPath**
FORMAware	- Style Manager; - Architecture Graph; - Architecture Management;	- Style Manager; - Architecture Management;	- Style Manager; - Architecture Graph; - Architecture Management;	- Style Manager; - Architecture Management;			**ADL**
Dynamic Adaptation with a DSL	- Exchange Behaviour; - Merge Behaviour;				Start Context	Stop Context	**DSL**
DSL for ATRON Robots	Role based	Role based	Role based	Role based	Role based	Role based	Role based

Table III
ADAPTIVE ACTION IN RECONFIGURATION TOOLS

adaptability of the clients that use it. However, the following considerations apply:

- the client is bound by the number of exposed services;
- the client cannot act on the characteristics of resources, because these are completely transparent to the client.

CLOUD computing extends the concept of service performed in REST, adding two more levels: in the first, the services are identified by a platform (PaaS: Platform-as-a-Service), and services are identified by a set of programs or libraries. In the second, the services are provided by an entire hardware infrastructure (IaaS - Infrastructure as a Service). Even in the case of the CLOUD, the characteristics of resources are not visible to the client, which is just a user resource consumption, and it is obviously ”limited” by the available services.

From Table III, it is clear how the management of the reconfiguration of a system, depending on an external or internal event, is often delegated to the architectural level. It is therefore significant the use of Architecture Description Language (ADL)[14] for modelling the system. In addition to architectural languages, we considered Domain Specific Languages (DSL)[15], which can model a particular domain or a particular technical solution. The combination of these two technologies allows to:

- navigate in a selective way the elements constituting the software architecture and;
- operate dynamically reconfiguration operations on the elements that constitute the system.

Differently from all the approaches here surveyed, PROTEUS introduces the concept of Virtual membrane and the corresponding action (virtual actions) to manage it at run time as a usual resource. These aspects make PROTEUS innovative and powerful.

VI. Conclusions and Future Work

In this article, we introduced PROTEUS. This language is characterized by the concept of virtual membranes, an abstraction that is designed to support the capacity of a system to meet the needs of an adaptation of its behavior, caused by an internal or external event.

Furthermore, we presented a simple application of PROTEUS, to show how the language can be used to implement reconfiguration plans. PROTEUS is still at an early stage of development, so it needs refinement and development. A formal semantics is currently being developed, in conjunction with a concrete implementation, to assess its ability to be used in different application contexts, its performance and its ability to scale in front of a large number of adaptations and its ease of use.

Concerning the implementation, we have already identified some tools to realize the concept of virtual membranes. In particular, we plan to use the SCALA language [16], and the concepts of traits [17] and class boxes [18].

Acknowledgment

This work has been supported by the EU-funded VISION ERC project (ERC-240555).

References

[1] H. T. Pu and Y. W. Wong, "User navigation behavior of a selective dissemination of web information service." in *iConference*, 2012, pp. 453–455.

[2] D. Vazhenin, "Cloud based web service for health 2.0." in *HCCE*, 2012, pp. 240–243.

[3] J. Dowling, T. Schäfer, V. Cahill, P. Haraszti, and B. Redmond, "Using reflection to support dynamic adaptation of system software: A case study driven evaluation," in *Proceedings of the 1st OOPSLA Workshop on Reflection and Software Engineering: Reflection and Software Engineering, Papers from OORaSE 1999*. London, UK, UK: Springer-Verlag, 2000, pp. 169–188. [Online]. Available: http://dl.acm.org/citation.cfm?id=646954.713478

[4] C. Ghezzi, M. Pradella, and G. Salvaneschi, "An evaluation of the adaptation capabilities in programming languages," in *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, ser. SEAMS '11. New York, NY, USA: ACM, 2011, pp. 50–59. [Online]. Available: http://doi.acm.org/10.1145/1988008.1988016

[5] L. Sitanayah, K. N. Brown, and C. J. Sreenan, "Fault-tolerant relay deployment based on length-constrained connectivity and rerouting centrality in wireless sensor networks." in *EWSN*, 2012, pp. 115–130.

[6] E. Ever, R. Luchmun, L. Mostarda, A. Navarra, and P. Shah, "UHEED - an unequal clustering algorithm for wireless sensor networks." in *Sensornets 2012*, 2012.

[7] A. Di Marco, F. Gallo, and F. Raimondi, "Proteus language," http://www.slrtool.org/proteus/index.php/Proteus, University of L'Aquila, Tech. Rep., 2012. Last access 04/05/2012, University of L'Aquila, Tech. Rep., 2012.

[8] P.-C. David and T. Ledoux, "Safe dynamic reconfigurations of fractal architectures with fscript," in *Proc. Fractal CBSE Workshop, ECOOP'06*, 2006.

[9] R. S. Moreira, G. S. Blair, and E. Carrapatoso, "FORMAware: Framework of reflective components for managing architecture adaptation." in *SEM*, 2002, pp. 115–129.

[10] S. Fritsch, A. Senart, and S. Clarke, "Addressing dynamic contextual adaptation with a domain-specific language," in *Proceedings of the 29th International Conference on Software Engineering Workshops*, ser. ICSEW '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 188–. [Online]. Available: http://dx.doi.org/10.1109/ICSEW.2007.26

[11] U. Schultz, D. Christensen, and K. Stoy, "A domain-specific language for programming self-reconfigurable robots," APGES 2007, Automatic Program Generation for Embedded Systems, 2007.

[12] R. T. Fielding, "Architectural styles and the design of network-based software architectures," University of California, IRVINE - 2000.

[13] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, U.S Department of Commerce - Special Publication 800-145, 2011.

[14] N. Medvidovic and R. N. Taylor, "A classification and comparison framework for software architecture description languages," *IEEE Trans. Softw. Eng.*, vol. 26, no. 1, pp. 70–93, Jan. 2000. [Online]. Available: http://dx.doi.org/10.1109/32.825767

[15] A. van Deursen, P. Klint, and J. Visser, "Domain-specific languages: an annotated bibliography," *SIGPLAN Not.*, vol. 35, no. 6, pp. 26–36, Jun. 2000. [Online]. Available: http://doi.acm.org/10.1145/352029.352035

[16] M. Odersky, "Scala language," online: http://www.scala-lang.org/.

[17] A. Bergel, S. Ducasse, O. Nierstrasz, and R. Wuyts, "Stateful traits and their formalization," Journal of Computer Languages, Systems and Structures, vol. 34, no. 2-3, 2008, pp. 83-108.

[18] S. Ducasse, "Supporting unanticipated changes with traits and classboxes," in *In Proceedings of Net.ObjectDays (NODE05*, 2005.

 ISBN: 978-1-61208-219-6